

**Títol:** Planificació, desenvolupament i proves del sistema de gestió logística i administrativa d'una empresa comercial

**Autor:** Jordi Soldevila i Corominas

**Data:** 27 de juny de 2014

**Director:** Miquel Barceló Garcia

**Departament:** Departament d'Enginyeria de Serveis i Sistemes d'Informació

**Titulació:** Enginyeria Informàtica

**Centre:** Facultat d'Informàtica de Barcelona (FIB)

**Universitat:** Universitat Politècnica de Catalunya (UPC)

SETELSYS	Memòria Projecte Final de Carrera	
----------	-----------------------------------	--

---

## DADES DEL PROJECTE

*Títol del Projecte: **Planificació, desenvolupament i proves del sistema de gestió logística i administrativa d'una empresa comercial***

*Nom de l'estudiant:* Jordi Soldevila i Corominas

*Titulació:* Enginyeria Informàtica

*Crèdits:* 37,5

*Director:* Miquel Barceló Garcia

*Departament:* Departament d'Enginyeria de Serveis i Sistemes d'Informació

---

## MEMBRES DEL TRIBUNAL *(nom i signatura)*

*President:* Xavier Burgués Illa

*Vocal:* Esteve Codina Sancho

*Secretari:* Miquel Barceló Garcia

---

## QUALIFICACIÓ

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---

## Taula de continguts

<b>1.</b>	<b>Introducció.....</b>	<b>9</b>
1.1	Motivació.....	9
1.2	Objectius.....	10
1.3	Organització de la memòria.....	12
<b>2.</b>	<b>Punt de partida .....</b>	<b>13</b>
2.1	L'empresa .....	13
2.2	Sistemes informàtics presents.....	14
2.2.1	Programari ERP principal.....	14
2.2.2	Subsistema d'aplicacions en Access.....	15
2.3	Problemàtica i necessitats derivades de la situació actual .....	17
2.3.1	Deficiències detectades a nivell arquitectònic o estructural .....	18
2.3.2	Deficiències detectades a nivell funcional .....	20
<b>3.</b>	<b>Decisions tècniques .....</b>	<b>21</b>
3.1	Selecció del software de desenvolupament i testing.....	21
3.1.1	Software de desenvolupament .....	21
3.1.2	Sistema gestor de bases de dades (SGBD) escollit .....	25
3.1.3	Software de testing .....	26
3.1.4	Treball en equip i desenvolupament compartit.....	29
3.2	Selecció de la plataforma hardware per allotjar l'aplicació.....	30
3.3	Arquitectura de la solució.....	32
3.4	Salvaguarda de les dades i del servidor .....	36
3.5	Convenis i nomenclatura utilitzada per a la codificació.....	38
3.5.1	Conveni global.....	38
3.5.2	Noms dels controls gràfics de les interfícies d'usuari (UI) .....	39
3.5.3	Nom de les classes.....	39
3.5.4	Nom de les propietats de les classes.....	40
3.5.5	Nom dels mètodes i funcions .....	40
3.5.6	Nom dels delegats, predicats, ... ..	40
3.5.7	Resta d'elements de desenvolupament.....	41
3.5.8	Funcions de test .....	41
<b>4.</b>	<b>Planificació.....</b>	<b>42</b>
4.1	Diagrama de Gantt inicial .....	42
4.2	Descripció de les tasques.....	43
4.2.1	Planificació i Control.....	43
4.2.2	Anàlisi de Requeriments.....	44
4.2.3	Especificació i Disseny .....	45

4.2.4	Desenvolupament .....	46
4.2.5	Testing .....	47
4.2.6	Finalització .....	48
4.2.7	Memòria del Projecte de Final de Carrera .....	48
4.3	<i>Projecte</i> .....	49
<b>5.</b>	<b>Anàlisi de requeriments</b> .....	<b>50</b>
5.1	<i>Inventari d'aplicacions satèl·lit</i> .....	50
5.1.1	Bloc 1: Logística .....	51
5.1.2	Bloc 2: Articles, estocs i inventaris .....	52
5.1.3	Bloc 3: Tarifes i preus .....	53
5.1.4	Bloc 4: CRM, consultes i altres .....	53
5.2	<i>Relacions de dependència entre els diferents blocs d'aplicacions</i> .....	54
5.2.1	Bloc 1: Logística .....	55
5.2.2	Bloc 2: Articles, estocs i inventaris .....	55
5.2.3	Bloc 3: Tarifes i preus .....	56
5.2.4	Bloc 4: CRM, consultes i altres .....	56
5.2.5	Conclusions sobre les dependències entre aplicacions .....	57
5.3	<i>Completesa i vigència de les aplicacions</i> .....	58
5.3.1	Bloc 1: Logística .....	58
5.3.2	Bloc 2: Articles, estocs i inventaris .....	59
5.3.3	Bloc 3: Tarifes i preus .....	59
5.3.4	Bloc 4: CRM, consultes i altres .....	59
5.3.5	Conclusions sobre la completesa i vigència de les aplicacions actuals .....	60
5.4	<i>Funcionalitat actual de les aplicacions vigents</i> .....	61
5.4.1	Aplicació "Pedidos" (Bloc 1: Logística) .....	61
5.4.2	Aplicació "Servir Todo Tiendas" (Bloc 1 – Logística) .....	66
5.4.3	Aplicació "Pendientes" i "Pendientes Y3" (Bloc 1 – Logística) .....	72
5.5	<i>Contrastar les funcionalitats vigents amb el client</i> .....	74
5.6	<i>Ampliació de funcionalitats per part del client</i> .....	75
5.6.1	Aplicació "Pedidos" (Bloc 1 – Logística) .....	75
5.6.2	"Servir Todo Tiendas" (Bloc 1 – Logística) .....	80
5.6.3	Aplicació "Pendientes" i "Pendientes Y3" (Bloc 1 – Logística) .....	84
5.7	<i>Informe previ i revisió de l'anàlisi de requeriments</i> .....	86
<b>6.</b>	<b>Especificació i disseny</b> .....	<b>87</b>
6.1	<i>Models de dades existents en el sistema</i> .....	87
6.1.1	Model de dades de l'ERP principal .....	87
6.1.2	Model de dades de la web comercial .....	94
6.2	<i>Especificació de l'arquitectura del sistema SetelSys</i> .....	99
6.2.1	Diagrama de components i dependències del sistema .....	99
6.2.2	Diagrama de classes del nucli del sistema (SetelSysCore) .....	102

6.3	<i>Especificació del gestor del sistema (SetelSysDataCom).....</i>	<i>104</i>
6.3.1	Casos d'ús.....	104
6.3.2	Diagrama associatiu de classes .....	109
6.4	<i>Disseny de la base de dades de SetelSys .....</i>	<i>110</i>
<b>7.</b>	<b>Desenvolupament .....</b>	<b>112</b>
7.1	<i>Metodologia utilitzada per al desenvolupament .....</i>	<i>112</i>
7.2	<i>Cronologia en el desenvolupament.....</i>	<i>113</i>
7.3	<i>Estructura de la solució .NET.....</i>	<i>113</i>
7.4	<i>Cicle de vida en la implementació de funcionalitats .....</i>	<i>114</i>
7.5	<i>Consideracions preses en el disseny de les UI i interacció amb l'usuari.....</i>	<i>115</i>
7.6	<i>Implementació de la base de dades principal de SetelSys .....</i>	<i>116</i>
7.7	<i>Implementació dels components del projecte.....</i>	<i>117</i>
7.7.1	SetelSysCore .....	117
7.7.2	SetelSysRunner.....	117
7.7.3	SetelSysWinService .....	118
7.7.4	<i>Maiden: Servei web.....</i>	<i>118</i>
7.7.5	<i>SetelSys: Aplicació web amb Silverlight.....</i>	<i>118</i>
7.8	<i>Implementació d'eines d'ajuda al desenvolupament i actualització a noves versions.</i>	<i>119</i>
7.9	<i>Llibreries i components externs utilitzats .....</i>	<i>119</i>
<b>8.</b>	<b>Testing .....</b>	<b>120</b>
<b>8.1</b>	<b><i>Introducció al Testing de Software .....</i></b>	<b><i>121</i></b>
8.1.1	Per què Testing? .....	121
8.1.2	Breu introducció història al testing de software.....	122
8.1.3	Tipus de proves.....	124
8.1.4	Nivells de proves.....	125
<b>8.2</b>	<b><i>Metodologia: Test Management Approach (TMap) .....</i></b>	<b><i>126</i></b>
8.2.1	Què és TMap? .....	127
8.2.2	Cicle de vida segons TMap .....	129
8.2.2.1	Planificació .....	130
8.2.2.2	Preparació .....	131
8.2.2.3	Especificació.....	132
8.2.2.4	Execució.....	135
8.2.2.5	Conclusió .....	138
8.2.2.6	Control .....	139
8.2.2.7	Infraestructura.....	140
<b>8.3</b>	<b><i>Aplicació de TMap en el Projecte .....</i></b>	<b><i>141</i></b>

<b>8.4</b>	<b><i>Fase de planificació</i></b>	<b>141</b>
8.4.1	Pla de Proves o Test Plan	141
8.4.2	Eina de gestió de proves – Fase de Planificació	144
<b>8.5</b>	<b><i>Fase de preparació</i></b>	<b>146</b>
8.5.1	Eina de gestió de proves – Fase de Preparació	146
<b>8.6</b>	<b><i>Fase d'especificació</i></b>	<b>149</b>
8.6.1	Eina de gestió de proves – Fase d'Especificació	150
<b>8.7</b>	<b><i>Fase d'execució</i></b>	<b>155</b>
8.7.1	Eina de gestió de proves – Fase d'Execució	156
8.7.2	Gestió d'errors	159
8.7.2.1	Eina de gestió de proves – Errors	159
8.7.2.2	Cicle de vida dels errors	161
8.7.2.3	Cicle nous casos de prova en cas d'error.	162
8.7.2.4	Alguns dels errors detectats durant les proves.	163
<b>8.8</b>	<b><i>Fase de conclusió</i></b>	<b>166</b>
<b>8.9</b>	<b><i>Fase de Infraestructura</i></b>	<b>167</b>
<b>8.10</b>	<b><i>Control</i></b>	<b>168</b>
<b>9.</b>	<b><i>Automatització</i></b>	<b>169</b>
<b>9.1</b>	<b><i>Introducció a les proves automatitzades</i></b>	<b>169</b>
<b>9.2</b>	<b><i>Estratègia de proves automàtiques i Eines</i></b>	<b>170</b>
9.2.1	Estratègia utilitzada	170
9.2.2	Eines utilitzades	171
<b>9.3</b>	<b><i>Patró de disseny de proves: Page Objects Pattern</i></b>	<b>172</b>
9.3.1	Introducció	172
9.3.2	Detalls de la implementació del Page Objects Pattern	173
<b>9.4</b>	<b><i>Implementació de les proves – Coded UI</i></b>	<b>174</b>
9.4.1	Estructura de la solució	174
9.4.2	Procés habitual	176
9.4.3	Elements de l'automatització i exemples	177
9.4.3.1	UI Maps	177
9.4.3.2	Automation.cs	179
9.4.3.3	Common Functions	180
9.4.3.4	Tests	184
<b>9.5</b>	<b><i>Implementació de les proves – Test Complete</i></b>	<b>187</b>
9.5.1	Automatització d'aplicacions en Silverlight	187
9.5.2	Opcions d'automatització de Test Complete	188
9.5.3	Keyword Test	189
9.5.4	Script Test	190
9.5.5	Combinació d'elements	192
<b>9.6</b>	<b><i>Conclusions sobre l'automatització</i></b>	<b>195</b>

<b>10.</b>	<b>Anàlisi econòmic .....</b>	<b>196</b>
<b>11.</b>	<b>Conclusions .....</b>	<b>198</b>
11.1	<i>Possibles millores futures .....</i>	<i>200</i>
<b>12.</b>	<b>Bibliografia i referències .....</b>	<b>201</b>



# 1. Introducció

## 1.1 Motivació

**L'objectiu principal d'aquest projecte és el de donar solució a una necessitat real en una empresa determinada, per a poder millorar i optimitzar els seus processos informatitzats.**

Dit així, pot semblar un resum fred i poc passional, però la veritat és que aquest projecte ens brinda la oportunitat d'analitzar la solució a un escenari prou complex, utilitzant eines i recursos coneguts, així com altres de nous necessaris per aconseguir donar a llum un projecte que requereix una àmplia gama de coneixements i característiques, amb la dificultat afegida de que no es parteix des de zero, sinó que l'empresa en qüestió ja disposa d'un sistema actualment en producció.

Com a motivació del projecte ens proposem un seguit de punts a potenciar i treballar:

- Aplicar les **tècniques** i l'**arquitectura** necessària per donar una solució modificable, adaptable i multi plataforma, acotant els costos de desenvolupament i proves, així com el de desplegament una vegada el sistema està en producció.
- Establir un nivell de **feedback** adequat amb el client final, per a focalitzar i determinar al màxim les necessitats reals desitjades al sistema.
- Assegurar una òptima **qualitat** en la solució proposada, garantint un alt nivell de cobertura de proves, i a la vegada, plantejant un conjunt de proves reutilitzables per a validar futures extensions del sistema, així com correccions que s'hagin de realitzar.
- **Planificar** l'anàlisi i desenvolupament del projecte tant acuradament com ens sigui possible, per tal d'obtenir un *timeline* el màxim real possible i minimitzar les desviacions, tant en temps com en recursos.
- Treballar amb un entorn **dependent** i **satèl·lit**. No partirem d'un escenari en blanc, on tota novetat gira entorn del nostre sistema, sinó que hem d'adaptar el nostre sistema a un altre de principal, i per tant controlar i planificar els canvis que es produeixen en ell per evitar problemes col·laterals en el nostre.
- Estudiar, analitzar i escollir les **eines** a utilitzar per construir les nostres idees i el nostre objectiu. Tant a nivell del desplegament, desenvolupament, verificació i testing, etc.

En l'apartat de conclusions podrem analitzar en profunditat el resultat de tot aquest plantejament, però seria sincer afirmar que tot projecte enriqueix i millora l'experiència, les eines i les metodologies aplicades en els anteriors. I aquest no n'és una excepció. Des dels primers anàlisis de requisits fins a la implantació final, no hem deixat d'aprendre i construir una base més ferma i experimentada pel futur, o si més no, això esperem.

## 1.2 Objectius

**SetelSys** pretén convertir-se en un motor de gestió de tots els procediments i aplicacions que actualment s'utilitzen en l'empresa per a donar resposta a les funcionalitats que no estan integrades en el seu sistema principal de gestió, al qual, d'ara en endavant ens referirem a ell com a ERP.

Tot i que actualment existeixen un seguit de programes que donen cabuda a aquestes necessitats (tal i com s'explica en l'apartat "[Punt de Partida](#)" d'aquesta memòria), aquests ho fan de manera precària, descontrolada, poc intuïtiva i eficient.

D'altra banda, SetelSys també haurà d'implementar la capacitat de gestionar processos automatitzats que s'executin de forma diària i parametritzada (tant pel que fa al calendari d'execució com el comportament), i de tot el subsistema necessari per a la gestió de la configuració de l'entorn.

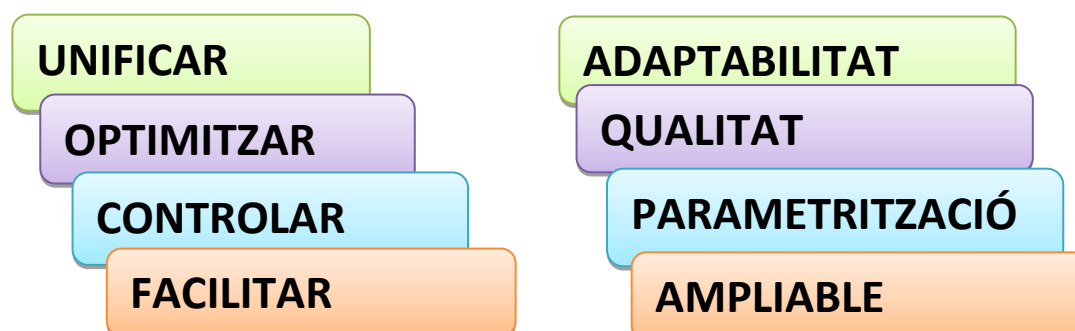
Podríem citar els següents objectius, clarament definits:

- Homogeneïtzar les diferents aplicacions i processos sota una mateixa aplicació. Volem un sistema capaç d'englobar i gestionar els diferents sistemes sota un mateix entorn de treball, i volem eliminar l'existència de solucions independents i disseminades en diferents equips servidors. Tot ha d'estar englobat en una sola solució i un únic servidor de dades i d'aplicacions.
- Permetre un creixement de les funcionalitats existents, així com facilitar la incorporació de noves funcionalitats, agilitzant-ne al màxim el procés, i a la vegada, fer-lo transparent a l'usuari. La incorporació de una nova funcionalitat o d'alguna modificació ha de suposar un cost mínim per a l'usuari.
- Coordinar i controlar les modificacions en les bases de dades de l'ERP, a través de les eines necessàries per a detectar possibles canvis en les bases de dades d'aquest que puguin involucrar problemes en el funcionament i la coherència de les funcionalitats del SetelSys.
- Dotar a l'aplicació d'un sistema de control i registre d'errors, per a poder monitoritzar l'estat del sistema en tot moment, així com poder generar resums diaris automàtics dels resultats de processos automatitzats que actualment no generen cap informació sobre el resultat de la seva execució.
- Control d'accés a les diferents funcionalitats de SetelSys a través d'un sistema d'usuaris propi que permeti definir les polítiques d'accés de cada usuari a cada funcionalitat, i inclús dependent de l'origen de la seva connexió.

- Agilitzar i facilitar la instal·lació i implantació de SetelSys en nous equips. La implantació i l'actualització de SetelSys ha de tenir un cost mínim en temps, complexitat i coneixements de l'usuari final.
- Millorar la qualitat dels processos i les funcionalitats, i això engloba elements com:
  - Assegurar la qualitat global del producte, amb l'objectiu de detectar els errors així que sigui possible, per tal de disminuir-ne el seu cost de correcció.
  - Validar que s'ha implementat una solució correcta en termes d'usabilitat, creant interfícies gràfiques intuïtives, manejables i clares.
  - Assegurar la solidesa de la solució, millorant la robustesa del sistema davant qualsevol tipus de situació imprevista, o ús incorrecte.
  - Implementar una solució amb rendiment correcte, que en permeti un ús fluid i ràpid.
  - Plantejar les proves de forma que es puguin reutilitzar al màxim a posteriori en noves versions de la solució, i a la vegada incorporant l'automatització de proves.
  - I en conjunt, realitzar un enfocament en metodològic i efectiu en termes de testing i qualitat
- Permetre administrar i parametritzar el comportament de les diferents funcionalitats de SetelSys perquè un usuari administrador pugui gestionar el comportament de les diferents funcionalitats sense dependre del desenvolupador. Es tracta d'obrir al màxim la personalització en aquells punts on detectem que és necessari.
- Reduir la dependència en equips de hardware que actualment estan obsolets i que de forma distribuïda allotgen programaris inconnexos i sovint de forma redundant i caòtica. Com més elements de hardware d'aquest tipus eliminem i aconseguim centralitzar la solució en un sol servidor, menys probabilitats hi haurà a l'empresa de paralització del treball per la caiguda d'un dispositiu de hardware.

Com podeu veure, no ens quedem curts en els objectius, però la veritat és que la solució desitjada i les necessitats del client així ho exigeixen. De fet, part d'aquest objectius sorgeixen directament de les necessitats detectades i transmeses per l'empresa, mentre que d'altres apareixen com a objectius derivats de les necessitats o prerequisits per a produir una solució completa i de qualitat, sota uns ulls crítics en l'anàlisi arquitectònic del sistema actual.

Resumint, les 8 paraules claus que sintetitzen els nostres objectius:



### 1.3 Organització de la memòria

El projecte de final de carrera “Setelsys” s’ha dut a terme de forma conjunta amb el meu company Marc Arnau i Borràs; vam optar per a realitzar el projecte conjuntament amb la idea de que fos un projecte més “real”, amb un component de treball en equip i distribuït, amb un afegit interès a nivell de planificació i coordinació.

S’ha optat per un treball conjunt en algunes àrees, però en altres hi ha hagut una clara separació de responsabilitats i tasques. I aquesta, s’ha fet precisament per a aprofitar millor els coneixements de cadascú dins el seu àmbit laboral. No obstant, aquesta separació no ha sigut total, en el sentit d’una caixa negra, sinó que s’ha fet amb un degoteig constant d’informació i coordinació, per a conèixer clarament en què consistia la feina de cadascú i veure com anava evolucionant el projecte.

A grans trets, a nivell conjunt s’han dut a terme les següents àrees:

- Planificació, decisions tècniques i coordinació
- Anàlisi de requeriments, dissenys funcionals
- Conclusions

I a nivell més individual, la fase de Desenvolupament s’ha dut a terme per part del Marc Arnau, i la fase de Testing per part del Jordi Soldevila.

En línia amb l’organització que s’ha mantingut durant el projecte, la memòria d’ambdós projectes s’ha realitzat de forma similiar, per a simplificar i facilitar la lectura i comprensió de com s’ha dut a terme el projecte, així com de la focalització que ha tingut cadascú en la seva part de la feina.

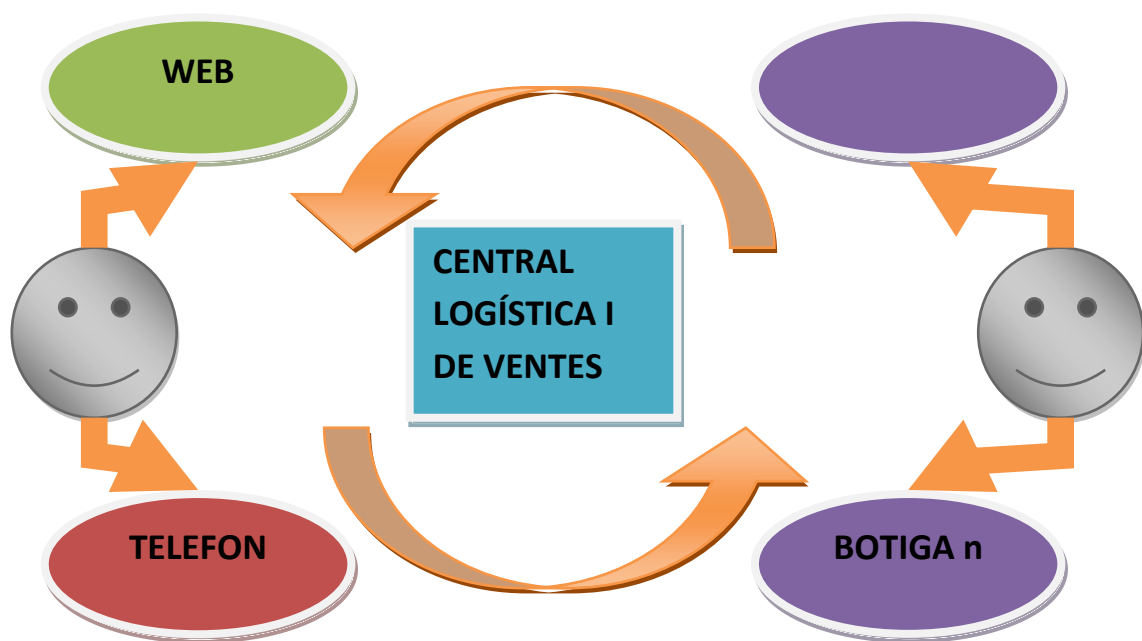
Les dues memòries tenen una part inicial i una part final similars, ja que tot el que és la planificació, anàlisi de requeriments i conclusions, entre altres, s’ha realitzat de forma conjunta. I el que són les àrees específiques del projecte, el seu responsable les ha explicat en detall en la seva memòria, i en la del company s’hi ha afegit un resum, per a mantenir una consistència i coherència, i permetre al lector de la memòria tenir una visió completa del projecte.

## 2. Punt de partida

Primer de tot i per poder entendre la situació inicial del sistema prèviament a la creació del projecte SetelSys, anem a definir els diferents elements i actors que intervenen en l'escenari:

### 2.1 L'empresa

Es dedica a la comercialització d'articles relacionats amb el món de les motocicletes, tant a través de botigues per la venda directa al client, com a través de la seva pàgina web i telecomandes a través de telèfon, tant a nivell nacional com internacional.



Actualment el **client** pot adquirir les seves compres:

- Directament a la **botiga**, en el cas de que disposi d'estoc.
- Fent la **reserva** a la botiga, a la qual s'enviarà el material des de la central.
- Fent la reserva a través de la pàgina **web**, amb la qual cosa s'enviarà directament al client, tot i que disposa de l'opció de recollir-ho en alguna de les botigues disponibles.
- Fent la reserva per **telèfon**, amb un comportament idèntic al d'una comanda via web.

## 2.2 Sistemes informàtics presents

### 2.2.1 Programari ERP principal

L'empresa disposa d'un sistema d'una tercera empresa, utilitzat per a la gestió de la major part de tasques tant a la central com a les botigues.

- **Gestió:** Documents de compra i venda d'articles, estocs, tarifes, magatzems, ...
- **Control de Presència** per al control i organització dels horaris dels seus treballadors.
- **Comptabilitat** vinculada amb la part de gestió.
- **Informes** i resums analítics tant de gestió com comptabilitat.
- Gestió de terminals de **punt de venda** (TPV)
- ...

Aquest ERP es divideix en dos aplicacions independents per cobrir les funcions esmentades:

- **Front:** Utilitzat com a programari de **Punt de Venta** a les botigues, amb funcionalitats limitades per aquest tipus d'activitat.
- **Manager:** Utilitzat com a programari de **gestió** des de la central, tant per la part de la gestió a nivell de compres i ventes, comptabilitat, gestió analítica, tarifes, traspassos entre magatzems, estocs, ...

Pel que fa a l'arquitectura d'aquests programaris, es caracteritza per:

- Bases de dades implementades en un servidor **Microsoft SQL Server**.
- Aplicacions **clients** desenvolupades per a executar-se sobre Microsoft **Windows**.

L'accés dels usuaris a aquestes dues aplicacions es fa a través de la pròpia **xarxa**, utilitzant:

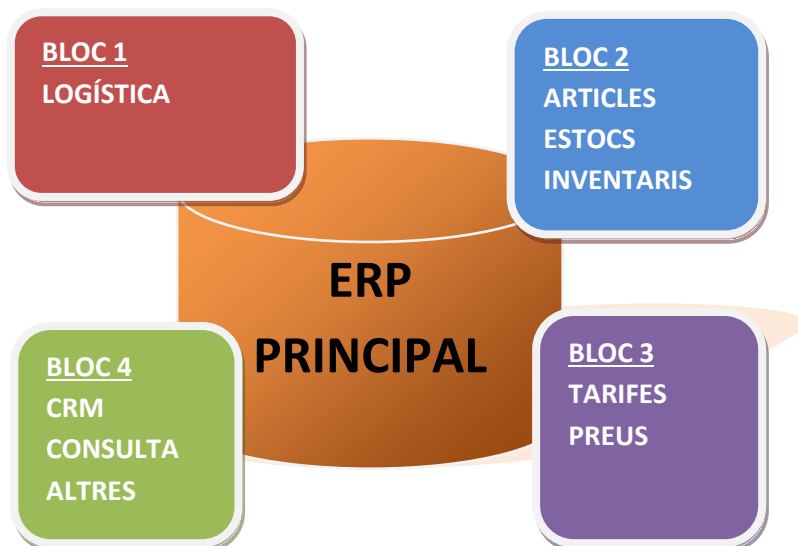
- Clients instal·lats **localment** en els equips dels usuaris.
- Accessos a través del servei de **Terminal Server** cap als servidors de dades on s'executen les aplicacions sobre la sessió en el servidors.

### 2.2.2 Subsistema d'aplicacions en Access

Existeix un subsistema, que s'executa de forma **paral·lela** al ERP principal, conformat per un gran nombre d'arxius de **Microsoft Access**, que treballen amb **vincles** a les bases de dades del ERP i utilitzen taules **auxiliars** de treball, generant arxius excessivament **extensos** i poc eficients, que cobreixen les **mancances** de l'ERP principal, tant en l'aspecte logístic, com el control d'articles, informes, etc.

La lògica ens diria que la forma més fàcil de resoldre aquestes mancances seria que el fabricant del software modifiqués el seu programari o el personalitzés pel client, però com que aquest es nega a procedir per aquesta via per política d'empresa, aquesta es veu obligada a dependre d'un sistema satèl·lit per cobrir les necessitats extres.

Aquests arxius o aplicacions d'Access impliquen un seguit de problemàtiques que analitzarem en el següent punt. De moment però analitzem com s'organitzen a nivell categòric o temàtic aquestes aplicacions, en base de l'àrea de l'empresa amb la que es relacionen.



#### *Bloc 1 – Logística*

Compren totes les aplicacions relacionades amb la gestió automatitzada de comandes. Per tal de poder agilitzar el procés d'enviament de comandes als clients (ja sigui per la seva recollida en botiga o bé per l'enviament directe al domicili del client), existeixen un seguit d'aplicacions per a generar els moviments de material entre magatzems o cap al client final, amb la generació dels corresponents albarans, regularització de l'estoc, moviments comptables, etc.

És un dels blocs més complexos, ja que l'algorísmica del procés té en compte molts factors i paràmetres.

SETELSYS	Memòria Projecte Final de Carrera	
----------	-----------------------------------	--

## ***Bloc 2 – Articles, estocs i inventaris***

Compren totes les aplicacions relacionades amb la gestió especialitzada d'articles (creació massiva, gestió d'articles especials, re càlcul de costos, ...), estocs (entrada massiva d'ubicacions d'articles, gestió de l'estoc dels proveïdors, mínims, ...) i també d'inventaris (gestió de regularitzacions).

## ***Bloc 3 – Tarifes i preus***

Compren les aplicacions necessàries per la modificació avançada de tarifes i preus associats als articles i la capacitat de generar còpies automàtiques i sota demanda dels preus. També existeix una aplicació per permetre la importació de preus a través d'arxius separats per comes o de Microsoft Excel.

Un dels elements més importants d'aquest bloc és un procés automatitzat encarregat de calcular i regenerar preus de tarifes derivades de tarifes base.

## ***Bloc 4 – CRM, Consulta i Altres***

Engloba aplicacions utilitzades per diferents tasques com la gestió de les targetes de fidelització, així com les consultes enregistrades que fan els clients finals o les pròpies botigues contactant amb la central.

En aquest bloc englobem també aplicacions puntuals i difícils de categoritzar que no tenen cabuda en cap altre bloc.



## 2.3 Problemàtica i necessitats derivades de la situació actual

En primer lloc cal entendre la situació des d'un global.

- Per una part tenim un sistema ERP principal, que dona cabuda a la majoria de necessitats de l'empresa, però està limitat pel que fa al creixement i a la especialització dels processos de l'empresa.
- Per una altra banda disposem d'un subsistema d'aplicacions de Microsoft Access que permet accomplir satisfactòriament els diferents processos interns del seu negoci, però presenta **deficiències** en bastants punts.

L'empresa té clar que el seu motor principal de gestió li funciona adequadament i com necessita, i vol enfocar les millores en la remodelació del sistema satèl·lit d'aplicacions que actualment presenten moltes **deficiències**. Vol construir un sistema satèl·lit robust, fiable i ampliable que cobreixi les necessitats actuals i permeti implementar amb certa "facilitat" necessitats futures.

Degut a la impossibilitat d'adaptar i personalitzar el programari del ERP Principal a les necessitats més concretes de l'empresa, aquesta requereix de la capacitat de disposar d'un sistema informàtic paral·lel i coordinat amb el sistema central actual per tal de poder dur a terme millores en els seus processos de treball que permetin optimitzar-los, millorar-los i unificar-los en una sola plataforma. I aquí, en aquest punt és on entra de ple el nostre projecte SetelSys.

### 2.3.1 Deficiències detectades a nivell arquitectònic o estructural

Existeixen un seguit de deficiències en el sistema satèl·lit actual, pel que fa a l'arquitectura de la distribució com del desenvolupament:

- **Limitació de l'arquitectura Microsoft Access a nivell de base de dades i entorn operatiu:** Sovint ens trobem davant d'aplicacions amb volums de dades suficientment grans com per desestabilitzar el funcionament d'aquest tipus de bases de dades o per ralentitzar-ne el funcionament fins a temps poc acceptables. D'altra banda, els enllaços a taules existents del sistema ERP principal es fan sempre amb connexions ODBC que sovint impliquen dependències en els canvis de les taules originals o de les credencials del domini d'Active Directory.
- **Processos poc eficients i poc segurs a nivell de coherència en cas d'errors:** En els aplicatius actuals no existeix la consideració d'errors, amb la qual cosa quan es produeixen en el procés d'una part del codi, els resultats són fins a cert punt indeterminats i "perillosos". Cal desenvolupar els aplicatius tenint en compte com actuar en cas d'error dins un procés.
- **Manca de control d'errors en temps d'execució:** No hi ha un tractament adequat utilitzant la filosofia *Try/Catch*. Qualsevol error es queda sense tractament. Tampoc es tenen en compte errors d'accessibilitat, de connexió amb la base de dades, violacions de claus primàries o foranies, ... Resumint: El control d'errors no existeix.
- **Distribució ineficient dels aplicatius:** Les diferents aplicacions no estan centralitzades sota un sol servidor, sinó que s'allotgen de manera distribuïda en diferents servidors (3-4 servidors). Això dificulta les tasques d'administració, salvaguarda de les dades, control d'execució, ... La fragmentació de les aplicacions suposa moltes pèrdues de temps a nivell administratiu i implica l'existència d'unes plataformes de hardware innecessàries, obsoletes i innecessàriament redundants.
- **Excés de dependències entre aplicatius:** Molts aplicatius presenten lligams entre ells a través de taules referenciades. Això vol dir que diferents arxius de Microsoft Accés depenen entre ells, creant sovint autèntiques teranyines de dependències. Un canvi en un dels arxius pot generar problemàtiques en molts arxius dependents o lligats a ell.
- **Cost elevat d'implementació de noves funcionalitats:** L'estructura actual d'aplicacions i les dependències anotades en l'anterior punt e) impliquen que la implementació de noves funcionalitats en un aplicatiu sigui molt elevat.
- **Limitació de l'entorn de desenvolupament:** Les eines de la plataforma Microsoft Access limiten les capacitats tecnològiques pel que fa al desenvolupament, tant pel que fa al llenguatge com a la capacitat d'integrar llibreries, elements externs, programació orientada a objectes, tecnologies més actuals i eficients, ...

- **Desconeixement de la vigència de les aplicacions:** Actualment existeixen un seguit d'aplicacions, que els propis usuaris desconeixen si se'n fa un ús o no. Cal determinar quines aplicacions són actuals i funcionals i quines no.
- **Aplicatius repetits per la mateixa funcionalitat:** Un altre element que indueix a la confusió és el fet que existeixin diferents aplicacions que donen la mateixa funcionalitat. Això en gran part és degut a la disseminació i falta de control en el desenvolupament d'aquest sistema satèl·lit.
- **Redundància de dades:** Existeixen aplicatius allotjats de forma repetida, no per una qüestió de balanceig ni de distribució organitzada de dades, sinó per una mala política de gestió d'aquests aplicatius i del sistema en general. Aquesta redundància descontrolada implica que sovint els usuaris treballin amb dades diferents en més d'un arxiu i generi un volum d'informació dividida, que indueix a errors o falta d'informació en molts casos.
- **Dades off-line:** Algunes aplicacions no consulten en línia les dades reals del sistema ERP principal, sinó que s'actualitzen un cop al dia a través d'un procés automatitzat. Això fa que certes consultes d'informació no estiguin actualitzades al moment i per tant pot provocar un desfàs informatiu respecte la realitat. Es vol aconseguir que el màxim d'aplicacions siguin amb connexió *on-line* amb les bases de dades principals del ERP.

### *En l'àmbit dels processos automàtics*

Durant les hores de inactivitat del sistema, s'aprofita per executar un seguit de processos automatitzats que poden generar un temps de còmput considerablement alt, i que no es poden executar durant els horaris de treball per tal d'evitar incidències en el rendiment del sistema. Aquests processos tenen aspectes a corregir **molt importants**:

- Optimitzar els temps de còmput, utilitzant consultes més optimitzades i ben estructurades a la Base de Dades.
- Controlar els errors en el procés i enregistrar-los en un diari d'incidències (log) per a consultes posteriors, anotant en tot moment el component, classe i rutina exacta on s'ha produït l'error conjuntament amb el missatge i codi d'error.
- Informar diàriament als administradors dels resultats de les execucions. Caldrà notificar a través de correu electrònic (parametritzant-ne els possibles destinataris), el resultat de l'execució diària dels processos automatitzats, indicant en tot moment si s'han completat correctament i en cas contrari indicant el missatge d'error.
- Dotar d'un sistema que permeti parametritzar quins processos es volen executar de forma automàtica i en quin interval concret (diari, múltiple diari, setmanal, mensual, sota demanda,...). S'haurà de facilitar el procés de configuració dels processos a través d'una interfície de gestió.

- Caldrà elaborar un sistema de màxima disponibilitat. Això vol dir que en cas d'un reinici del servidor, el sistema d'execució de processos automatitzats s'iniciï automàticament a l'iniciar el sistema operatiu on està allotjat i no depengui d'un factor humà per la seva reactivació.

### 2.3.2 Deficiències detectades a nivell funcional

A part de les deficiències a nivell estructural o arquitectònic, existeixen unes mancances actuals a nivell funcional dels aplicatius existents, tant pel que fa a noves necessitats no implementades com la millora de les ja existents. Les podem englobar en els diferents blocs que em subdividit anteriorment:

#### En el "Bloc 1" (Logística)

- Optimitzar el procés d'enviament de les comandes directament a client, generant els corresponents moviments d'estocs entre les botigues i la central per agilitzar-ne el temps d'entrega.
- Millorar el sistema de reserves en les botigues per assegurar un immediat moviment d'estocs per a cobrir les reserves dels clients.
- Evitar moviments d'estocs innecessaris.
- Ajudar en el procés d'organització logística dins del magatzem central. Gestió de caixes i seguiment de caixes de reserva.
- Permetre la proposta de traspassos entre botigues i comandes a proveïdor, basant-se en categories d'articles i les necessitats que porten explícites per cada magatzem per tal d'automatitzar el procés.

#### En el "Bloc 2" (Articles, estocs i inventaris)

- Agilitzar el procés de creació d'articles, amb les seves talles i colors, amb una inicialització ràpida de les tarifes per defecte i estocs mínims i màxims pels diferents magatzems.
- Sistema de gestió de les faltes d'estoc en els diferents magatzems per tal de poder ajudar en els processos de regularització d'estoc i no computar estocs erronis en els moviments de material entre les diferents botigues i la central.
- Automatitzar la captura d'ubicacions dels articles al magatzem central, que actualment requereix un treball molt manual.
- Permetre el recàlcul dels costos dels articles per ajustar el seu cost d'estoc, en base dels diferents documents de compra que existeix en l'històric de la base de dades.

#### En el "Bloc 3" (Tarifes i preus)

- Permetre generar còpies manuals i automàtiques de tarifes de preus de forma que un usuari pugui accedir a la còpia i restauració amb una interfície senzilla i fàcil d'utilitzar.
- Parametritzar el procés de regeneració de preus i tarifes d'articles, de forma que es puguin escollir quines tarifes es volen regenerar en base de les tarifes bàsiques.

#### En el "Bloc 4" (CRM, Consulta i Altres)

- Dotar d'un sistema potent de consulta de comandes pendents de servir al client, amb la màxima informació sobre el seu estat present i futur.

### 3. Decisions tècniques

#### 3.1 Selecció del software de desenvolupament i testing

##### 3.1.1 Software de desenvolupament

Tenint en compte les següents premisses pel que fa als requisits de la solució a desenvolupar i implementar:

- *Solució multi plataforma i portable*: existeixen equips clients amb diferents sistemes operatius.
- *Actualització ràpida i àgil del sistema per a tots els clients*: les actualitzacions s'han de desplegar amb la màxima transparència i la mínima intervenció tècnica i de l'usuari.
- *Ràpid desplegament en nous equips clients*: poder instal·lar amb la màxima facilitat la solució minimitzant els costos administratius.
- *Disposar de les eines necessàries per a crear una aplicació prou complexa com la de SetelSys*.
- *Afinitat en l'ús de les eines*: no hem de sacrificar cap potencial ni possibilitat en l'elecció, però en igualtat de condicions, és important poder desenvolupar l'aplicació amb una eina més coneguda i familiar.

El primer plantejament que cal fer-nos és l'avaluació sobre si la solució ha de ser una aplicació d'escriptori o bé una solució basada en la web. D'aquest plantejament en surt el primer anàlisi d'avantatges i inconvenients:

APLICACIÓ D'ESCRITORI (Client local)	
Avantatges (+)	Inconvenients (-)
➤ Menys càrrega en els servidors	➤ Major requeriment de maquinari en els equips clients.
➤ Desenvolupament més ràpid	➤ Implantació més lenta
➤ Interfície més potent i fàcil de personalitzar.	➤ Dependència de plataforma (sistema operatiu i versió)
	➤ En cas d'haver d'aplicar una nova actualització és necessari instal·lar-la en cada equip de la xarxa o desenvolupar un sistema d'actualitzacions automàtiques.

APLICACIÓ WEB	
Avantatges (+)	Inconvenients (-)
➤ No requereix una instal·lació assistida. Simplement tenir un navegador web actualitzat.	➤ Necessari verificar compatibilitat amb tots els navegadors (o si més no els usats en l'empresa)
➤ Implantació immediata	➤ Requereix d'un SERVIDOR WEB físic, encarregat de donar servei a la INTRANET.
➤ Portabilitat màxima (independència de plataforma i de versió)	
➤ Menys tràfic de dades, solament s'envia la informació final al client, la resta s'executa en el servidor a través de la xarxa local.	
➤ Actualitzacions automàtiques, sense necessitat de distribuir.	
➤ Major coneixement i especialització en la plataforma de desenvolupament.	

En base a les avantatges i inconvenients presentats, ens vàrem decantar per seleccionar com a plataforma de l'aplicació el treballar amb una **APLICACIÓ WEB**, principalment pels següents motius:

- Compleix les tres primeres premisses que hem anotat com a requisits de les eines a utilitzar: multi plataforma, fàcil actualització i ràpid desplegament.
- Ambdós opcions compleixen les dues últimes premisses anotades (potencial i afinitat). No obstant estem més familiaritzats amb el desenvolupament d'aplicacions d'escriptori. Evidentment, les premisses ens marquen el camí clarament cap a l'opció de desenvolupar una aplicació web en forma d'intranet.

Una vegada decidida la plataforma en que distribuïrem i mostrarem l'aplicació de **SetelSys**, una intranet web privada, cal decidir quina tecnologia utilitzem de les existents. Les opcions que apareixen en la primera ronda de selecció són:

- Desenvolupament utilitzant **ASP .NET** a través d'una arquitectura MVC (Model – Vista – Controlador).
- Desenvolupament utilitzant Microsoft **Silverlight** a través d'una arquitectura MVVM (Model - Vista – Vista - Model).
- Desenvolupament utilitzant altres alternatives, com podrien ser l'ús combinat de **PHP + jQuery**.

D'aquesta primera ronda eliminem la tercera opció, bàsicament pels següents motius:

- Menys familiaritzats amb aquestes plataformes, i per tant el cost d'aprenentatge i formació seria molt elevat.
- Les bases de dades principals del sistema estan implementades sobre Microsoft SQL Server, i tot i que aquesta situació no implica forçosament treballar amb eines de desenvolupament Microsoft, som conscients de les eines d'integració que existeixen en la suite del Microsoft Visual Studio pel treball amb bases de dades i components del mateix fabricant.
- No sacrifiquem cap potencial ni limitem les possibilitats. L'entorn de desenvolupament de Visual Studio ens brinda totes les eines necessàries per a la implementació satisfactòria del projecte SetelSys.
  - Programació orientada a objectes i sobretot a events.
  - Comunicació amb serveis web SOAP, WCF, RIA.
  - Comunicació amb bases de dades.
  - Creació de llibreries DLL
  - Creació de serveis per entorn Windows (WinServices)
  - Gestió adequada del codi i de l'entorn de depuració:
    - Tasques pendents (ToDo Lists)
    - Punts de parada (BreakPoints)
    - Gestió de marcadors
    - ...
  - I altres que no seran necessàries per aquest projecte.

Finalment hem arribat a una disjunció entre dos possibles plataformes:

- ASP .NET <sup>1</sup> (Arquitectura **MVC**) <sup>2</sup>
- Silverlight <sup>3</sup> (Arquitectura **MVVM**) <sup>4</sup>

L'escollida entre aquestes dos és el **Silverlight**, pels següents motius:

- Més orientada a l'elaboració d'una **Intranet privada**, que no pas la plataforma ASP .NET. Aquesta última està més pensada per solucions públiques que requereixin cert nivell d'obertura i publicació de la seva meta informació (sobretot pel posicionament web).

---

<sup>1</sup> Microsoft – [The official Microsoft ASP.NET site](#)

<sup>2</sup> Microsoft – [ASP MVC Overview](#)

<sup>3</sup> Microsoft – [Silverlight](#)

<sup>4</sup> MSDN – [Design Patterns - MVVM - Model View ViewModel Pattern](#)

- Permet l'elaboració **d'interfícies d'usuari més riques** i basades en els controls WPF (*Windows Presentation Foundation* <sup>5</sup>), que donen molta més gama de controls i personalització dels mateixos. Per una intranet corporativa on hem de mostrar moltes dades, graelles, gràfics representatius, etc., necessitem d'utilitzar una interfície rica i potent. És el que Microsoft anomena "*Reach vs. Rich*"<sup>6</sup>.
- Aquesta aproximació de Silverlight en l'extrem "*Rich*", ens permet desenvolupar amb molta més **agilitat i rapidesa** les **interfícies** d'usuari, sense dependre de components de terceres empreses que si que hauríem d'adquirir per aconseguir els mateixos resultats amb ASP .NET
- ASP .NET és una plataforma "*server side*", mentre que **Silverlight** és "*client side*". Això vol dir que en ASP .NET tot s'executa sobre el servidor en tot moment, mentre que el client Silverlight ho fa en el client. El gran avantatge d'aquesta característica és que, tot i tenir una càrrega inicial un pel més lenta, una vegada carregat l'script en la màquina client, la transició entre pàgines, finestres, etc., és molt més **fluida**. I si a més tenim l'aplicació guardada en la memòria cau del nostre ordinador, aleshores l'únic element depenent de la connexió seran les crides als serveis web de consulta de dades remotes.
- Estem més **familiaritzats** també en solucions basades en WPF que no en les de ASP.
- Tal i com s'explicarà en l'apartat corresponent a les eines seleccionades per a les fases de proves, la selecció del Silverlight pot comportar una sèrie de inconvenients menors a nivell d'automatització. En aquest cas, per a poder usar directament les eines d'automatització de proves que ofereix la *suite* del Visual Studio, requeriria una sèrie de modificacions en els objectes de l'aplicació. No obstant això, s'han avaluat les alternatives, optant per una solució que permet usar el Silverlight sense perdre potencial a nivell d'automatització.

---

<sup>5</sup> Microsoft Developer Network – [Windows Presentation Foundation](#)

<sup>6</sup> Tech Republic – [Create rich interfaces with Microsoft Silverlight](#)



### 3.1.2 Sistema gestor de bases de dades (SGBD) escollit

L'ERP principal disposa actualment ja d'un clúster de servidors dedicats a allotjar les aplicacions i bases de dades pròpies del seu sistema utilitzant Microsoft SQL Server com a SGBD. Pel nostre projecte SetelSys, utilitzarem també Microsoft SQL Server. En concret utilitzarem la versió **Microsoft SQL Server 2008 R2 Standard**.

#### Per què Microsoft SQL Server?

Els principals motius de la selecció del Microsoft SQL Server són les següents:

- Suficientment potent per a integrar les dades del projecte SetelSys.
- Integració més ràpida i fàcil amb l'entorn de Visual Studio .NET.
- Més facilitats a l'hora d'interactuar amb les altres bases de dades ja existents en el sistema i basades també en Microsoft SQL Server.
- Disposa de les eines necessàries per a una correcta administració de tasques programades de manteniment i/o de còpia de seguretat (SQL Server Agent).
- Només caldrà desenvolupar un component de connexió amb un motor de SGBD. El mateix que construïrem per les connexions amb la base de dades del ERP ens serviran per la base de dades de SetelSys.

#### Per què l'edició "Standard"?

De les diferents versions existents (DataCenter, Enterprise, Standard, Web, Workgroup, Express with Advanced Services, Express with Tools, Express), i sent conscients de que existeixen moltes d'altres categories on podríem analitzar les diferències entre versions, hem optat per la versió "Standard" pels següents motius, que considerem més determinants:

- **Escalabilitat:** A nivell d'escalabilitat ens permet fer ús de fins a 4 CPUs, utilitzar un màxim de 64 GB de memòria operativa, i un màxim de 524 PBs de memòria d'emmagatzematge. Altres versions de menors prestacions limiten possibles vies de creixement posteriors, sobretot limitant la memòria operativa disponible o l'espai d'emmagatzematge.
- **Disponibilitat:** Permet una configuració posterior en clúster, mentre que la resta de versions menors, no disposen d'aquesta funcionalitat. Si el client finalment optés per la configuració d'un clúster, podríem seguir utilitzant la mateixa versió de SQL Server.
- **Analysis Services:** La versió "Standard" disposa de l'eina SQL Server Analysis Services, amb eines per l'anàlisi actiu del servidor (nivell de càrrega, temps de còmput de les consultes, ...). La resta de versions menors, no la incorporen.

### 3.1.3 Software de testing

En referència al software que s'ha seleccionat per a dur a terme totes les tasques necessàries a nivell de testing, cal tenir en compte principalment quatre grans àrees de treball:

- **Gestió de les proves:** eines necessàries per a gestionar tot el que fa referència a les proves: Pla de proves, Especificació de les proves, gestió de l'execució i resultats, així com la posterior gestió dels errors que s'han localitzat.
- **Automatització:** eines usades per a la posterior automatització de les proves seleccionades, tenint en compte tant l'aplicació web com el *backend*, i les tecnologies seleccionades per a implementar-los.
- **Repositori de codi:** en aquest cas, i tal i com es veurà en el següent apartat, s'ha usat tant durant el desenvolupament com durant les proves les mateixes eines de control.

#### *Gestió de les proves:*

Actualment en el mercat podem trobar una gran varietat de programari de “*Test Management*”, que ens permeten una gestió integral de les proves. Les eines de gestió de proves s'usen per a guardar la informació sobre **com** es duran a terme les proves, planificar totes les activitats de testing, així poder obtenir informació constant i fidedigna de com evolucionen les proves.

Cadascuna de les eines aporta una sèrie de capacitats i funcionalitats, però en el nostre cas l'eina a seleccionar havia de complir els següents requeriments:

- Permetre'ns gestionar des del principi fins al final els nostres requeriments, proves i fases d'execució.
- Tenir una clara traçabilitat de tots els elements del projecte (ex: poder veure d'un requisit concret, quin és l'estat de les proves i quines incidències relacionades s'han detectat).
- Una integració senzilla amb la resta d'eines seleccionades per al desenvolupament
- I finalment, i tot i no ser un requisit indispensable inicialment, havia de permetre treballar de forma vinculada a nivell de proves automàtiques, amb les eines que es seleccionessin.

El fet que a nivell de desenvolupament haguéssim optat per a treballar amb la *suite* del Microsoft Visual Studio, ens va simplificar de forma clara la selecció de l'eina de gestió de proves. Hi ha diverses alternatives força conegudes amb les seves corresponents capacitats, però no obstant, nosaltres vam optar des del primer moment per utilitzar el **Microsoft Test Management (MTM)**, utilitzant a la vegada el **Team Foundation Server (TFS)**, bàsicament perquè el propi MTM el requereix per a desar tots els elements.

### *Automatització de proves:*

Com s'ha mencionat breument en l'apartat de selecció de les eines i tecnologies de desenvolupament, aquesta selecció té una influència directa sobre la selecció de les eines per a l'automatització de les proves.

El punt més important ha tenir en compte és el fet, que amb la selecció de Silverlight per al desenvolupament de part del nostre sistema, vam haver de prendre algunes decisions per a poder realitzar correctament l'automatització.

Una de les nostres primeres premisses era que les eines d'automatització s'integressin correctament i fàcilment amb la resta d'eines de desenvolupament, fet que ens va portar ràpidament a utilitzar el mateix Visual Studio, concretament els que anomenen **Coded UI Tests**, tests automàtics per a realitzar proves usant la seva interfície, és a dir, proves funcionals que utilitzen de forma directa tots els elements del sistema, tal i com veurem més en detall en l'apartat específic de [proves automàtiques](#).

Per a la part del *backend* s'ha optat directament per aquesta solució, ja que s'integrava correctament i sense problemes.

No obstant això, segons la pròpia documentació de Microsoft, per a poder reconèixer i interactuar amb els diferents objectes de l'aplicació codificada usant Silverlight, cal realitzar primer de tot una sèrie de modificacions en el codi:

- Caldria afegir la referència  
Microsoft.VisualStudio.TestTools.UITest.Extension.SilverlightUIAutomationHelper.dll a l'aplicació per tal de que els controls Silverlight es poguessin identificar
- A més, aquesta *dll* caldria afegir-la de forma condicional i no distribuir-la amb la resta de la solució.<sup>7</sup>
- I amb tots aquests canvis, en molts dels casos, es segueixen tenint dificultats a l'hora de poder reproduir correctament les proves.

Degut a aquestes circumstàncies, vam optar per utilitzar una eina diferent per al *frontend*, una eina que per experiències laborals anteriors requeria de molt pocs passos per a poder detectar els objectes de Silverlight senzillament. L'eina seleccionada va el **TestComplete**, de l'empresa SmartBear. Test Complete és una eina purament d'automatització, per a gestionar la creació, el manteniment i la posterior execució de les proves, tant per aplicacions d'escriptori, web com mòbil. En l'apartat corresponent veurem en detall el funcionament d'aquesta aplicació, i les proves automatitzades, però per a poder interactuar amb l'aplicació únicament ens calia realitzar un *patch* del fitxer .xap en el servidor, i d'aquesta manera l'aplicació ja era capaç de reconèixer i interactuar amb els objectes.

---

<sup>7</sup> [Testing Silverlight Applications with Coded UI Tests or Action Recordings](#)

La gràcia de seleccionar aquesta eina és que, a part de la facilitat amb la que ens permet interactuar amb l'aplicació, té una integració molt bona amb el Team Foundation Server i la suite de Visual Studio, fet que ens ha permès lligar correctament les proves dissenyades al Test Manager, amb les seves corresponents proves automàtiques.<sup>8</sup>

### *Repositori de codi:*

Sense necessitat d'entrar en molt de detall, i tal i com s'explica a continuació, a nivell de testing també s'ha usat les mateixes eines de control de versions per a gestionar el codi desenvolupat durant la fase de proves automàtiques.

---

<sup>8</sup> SmartBear – [Integration With Microsoft Visual Studio](#)

### 3.1.4 Treball en equip i desenvolupament compartit

Per tal de poder realitzar conjuntament les tasques de desenvolupament, així com les de les fases de proves, i durant la correcció d'errors, hem fet ús d'una eina de control de versions, utilitzant un client de l'**Apache Subversion (SVN)** <sup>9</sup>, anomenat **TortoiseSVN**. <sup>10</sup>

Aquesta aplicació ens permet sincronitzar el codi de forma remota i controlar els canvis generats en cada nova versió que es penja al **repositori** SVN. Els motius de l'elecció de TortoiseSVN:

- Ens permet integrar un **servidor Apache propi** en un equip servidor intern a la nostra xarxa de treball i amb accés remot a través de les corresponents configuracions, per a poder disposar d'una **eina gratuïta** (ja sigui per un ús comercial o no) sense cap cost per nosaltres.
- Disposa de les **funcionalitats** necessàries pel nostre projecte:
  - Descarregar i mantenir actualitzada una rèplica del repositori en el nostre equip.
  - Integració amb l'explorador de Windows (ràpidament podem visualitzar si un arxiu està actualitzat a la última versió o no a través de la icona de l'arxiu).
  - Capacitat d'explorar el contingut del repositori.
  - Ens permet descarregar registres de tots els canvis existents abans de pujar una nova versió al repositori.
  - Ràpida i còmoda actualització dels nostres arxius locals, ja sigui per descarregar la última versió (Update) o bé per a penjar tots els canvis que hem fet (Checkout).
- A través del complement **AnkhSVN** <sup>11</sup>, que instal·lem dins l'entorn del Visual Studio, podem visualitzar tota la informació referent al sistema de subversions, i per tant tenim accés constant i immediat a la seva informació (última versió, última data i hora de modificació, ...)

---

<sup>9</sup> [Apache Subversion](#)

<sup>10</sup> Tortoise SVN - [Home](#)

<sup>11</sup> AnkhSVN – [Subversion Support for Visual Studio](#)

### 3.2 Selecció de la plataforma hardware per allotjar l'aplicació

Per poder evitar l'ús de varis servidors amb processos i funcionalitats disperses, i no sobresaturar els servidors dedicats exclusivament a les connexions dels terminals i accés a les bases de dades de l'ERP Principal és necessari un nou servidor que ens permeti:

- Unificar els servidors actuals (justos de recursos i obsolets a nivell de plataforma).
- S'especialitzi i es dediqui a servir peticions del sistema satèl·lit del SetelSys, ja sigui a través de la seva aplicació web hostatjada en un servidor IIS o bé per l'aplicació de manteniment del sistema.
- Permeti una execució controlada i autònoma dels processos automatitzats necessaris.
- Disposi de l'estabilitat i fiabilitat necessària, així com de la potència suficient per a dur a terme aquestes múltiples tasques de manera eficient i paral·lela.




Actualment l'empresa disposa d'una sala de servidors amb un armari on resideixen els diferents elements centrals del sistema hardware (servidors, SAIs, commutadors de xarxa, routers,...). Aprofitarem aquesta estructura i eix neuràlgic per integrar el servidor dedicat a les tasques del sistema SetelSys, en format de rack 2U, per poder-lo connectar també amb el sistema d'alimentació ininterrompuda i deixar-lo integrat en l'armari principal.

Plantegem doncs l'adquisició del següent servidor i components necessaris per la seva posada en marxa i operativitat:

<p><b>x1</b></p> <p>Servidor HP ProLiant DL360P Gen8          Processador Xeon E5-2620 a 2.00GHz          8 GB de memòria RAM ECC          Controladora SAS HP Smart Array P420i          DVD-RW          Format rack 2U</p>	
<p><b>x3</b></p> <p>Discs durs de 500 GB Gen8 6GB SAS          HotPlug (connexió en calent)          15.000rpm          2.5"          Configurats en RAID 5</p>	
<p><b>x1</b></p> <p>Microsoft Windows Server 2008          Versió "Standar".</p>	
<p><b>x5</b></p> <p>Llicències d'accés de client (CAL)          Incorporades amb el Windows Server 2008.</p>	

SETELSYS	Memòria Projecte Final de Carrera	
----------	-----------------------------------	--

El client també ha optat per incorporar els següents serveis i components opcionals oferts:

<b>x1</b> Extensió de garantia HP CarePack 3 anys de garantia amb resposta de servei l'endemà de l'obertura de la incidència.	
<b>x1</b> Mòdul extra de memòria RAM ECC de 8 GB El sistema queda aleshores amb 16 GB de memòria RAM total.	
<b>x1</b> Font d'alimentació redundant HP Permet mantenir l'equip en funcionament en cas de la fallida de la font d'alimentació principal.	

### *Possibles millores descartades en la plataforma de hardware*

També s'ha plantejat al client l'opció d'instal·lar dos servidors amb la mateixa configuració que la mencionada en les anteriors taules amb un sistema de "clúster". Una configuració en "clúster" ens permetria que el sistema reaccionés transparentment si es produeix la caiguda d'un dels servidors.

En el moment que un dels servidors queda fora de servei pel motiu que sigui, l'altre servidor "redundant" del clúster, s'encarrega de la gestió del sistema i per tant evita la paralització dels processos de l'empresa.

No obstant aquesta opció ha estat desestimada inicialment pels costos duplicats a nivell de hardware que això suposa.

### 3.3 Arquitectura de la solució

Per tal de poder desplegar de forma eficient i controlada el sistema de gestió del SetelSys, el sistema estarà conformat per:

- **SetelSys Server:** Equip servidor, encarregat d'allotjar les aplicacions, serveis web, serveis de Windows i les bases de dades pròpies del SetelSys, amb la configuració de hardware anotada en l'anterior apartat.
- **Internet Information Services (IIS)** instal·lat i configurat com a rol en dit servidor i encarregat de subministrar i implementar l'accés a l'aplicació web de SetelSys, així com als serveis web necessaris per la interacció amb les bases de dades.
- **SetelSys:** Aplicació web “*Front End*” desenvolupada amb Silverlight v4: Es podria haver desenvolupat amb Silverlight 5, però degut a que certs equips clients treballen amb sistemes operatius bastant antics, i que la versió Silverlight té alguns problemes de compatibilitat amb aquestes versions, hem optat per desenvolupar-ho amb la SDK v4 del Silverlight.
- **SetelSys Core:** Nucli intern de la solució, encapsulat en una llibreria de Windows (DLL), amb totes les funcionalitats necessàries per l'execució de la solució:
  - Capa de gestió i control d'execució i registre d'errors (Midgard)
  - Classes d'accés als diferents entorns de dades:
    - WebDataCom
    - SetelSysDataCom
    - ERPDataCom
  - Classe per l'accés controlat a bases de dades SQL Server, ja sigui en execucions simples o de múltiples comandes, i també poden escollir l'execució dins de transaccions.
  - I la resta de classes que analitzarem més en profunditat en l'apartat del desenvolupament.
- **SetelSys Crono:** Servei de Microsoft Windows encarregat d'executar tots els processos automatitzats segons la planificació especificada en la seva configuració.
- **Maiden:** Servei web WCF per a resoldre peticions dels clients de l'aplicació web SetelSys i el servidor de l'aplicació web que a la vegada actua com a servidor de bases de dades.
- **SetelSys Runner:** Programari client “*Back End*” desenvolupat amb WinForms utilitzant la versió 4 de .NET Framework. Aquest programari s'utilitza per operacions de manteniment i control bàsiques, tant pel que fa als paràmetres del sistema com a l'observació de l'execució de processos automatitzats i la configuració de la seva planificació, entre d'altres funcionalitats que veurem més endavant.



Cal anotar que el servidor web de l'aplicació està hostatjat en la mateixa sala de servidors des d'on té accés per LAN a les diferents bases de dades allotjades en els servidors de l'ERP principal.

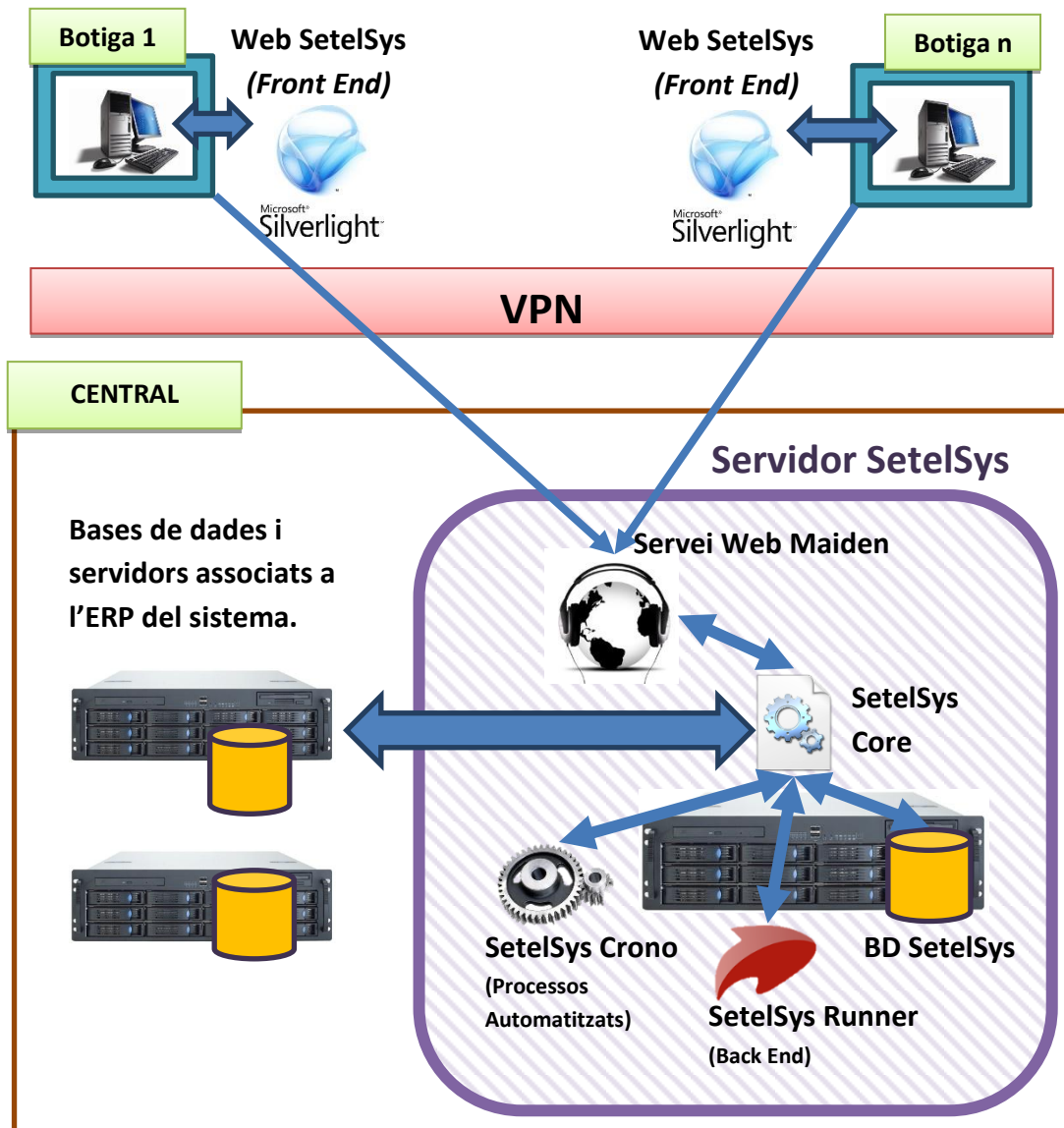
Totes les connexions remotes que s'estableixen cap al servidor de l'aplicació web del SetelSys són des de les diferents botigues cap a la central (on resideixen els servidors) a través d'una VPN establerta per hardware amb encaminadors Cisco, utilitzant una xarxa NetLan de Movistar.

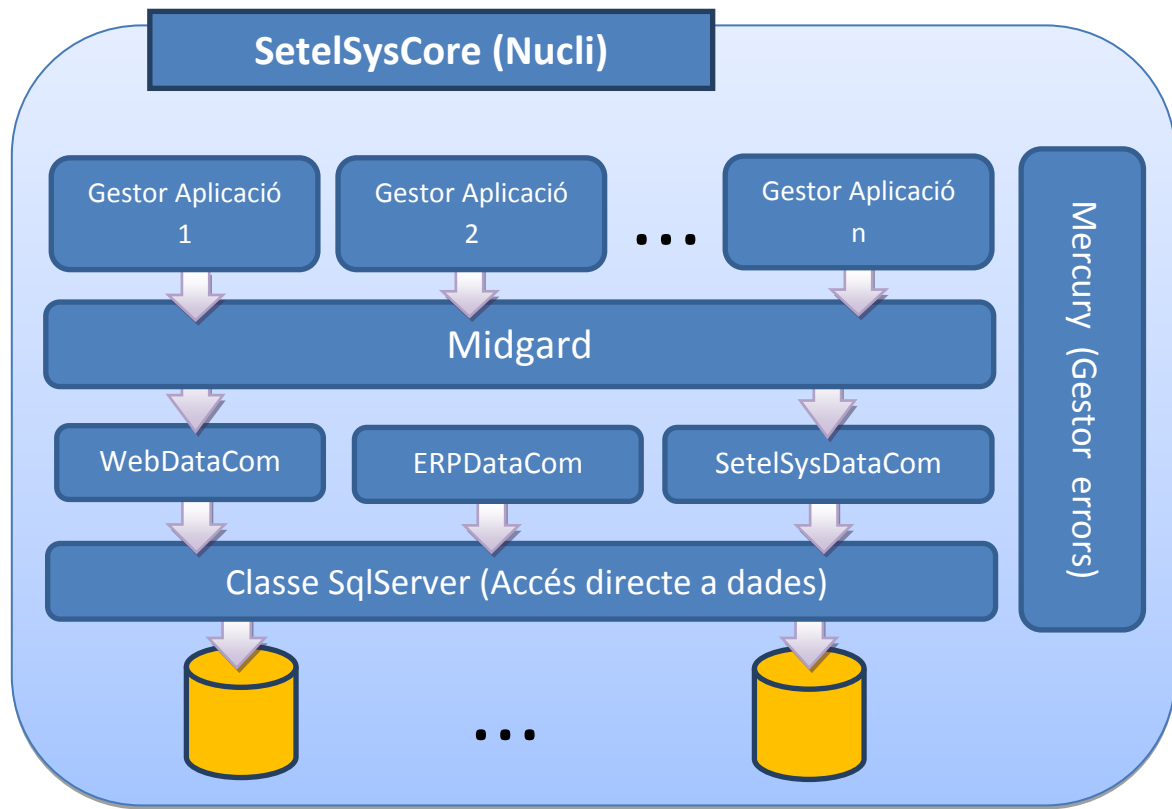
A través d'aquesta VPN les dades viatgen sempre sobre un canal segur i encriptat, fent innecessària la implementació d'un certificat SSL per al desplegament de l'aplicació web del SetelSys.

També s'han creat certes aplicacions d'ajuda en el control del sistema i per a facilitar la instal·lació d'actualitzacions:

- **DB Comparer** → Comparador d'esquemes de bases de dades.
- **FTP Deployer** → Per la instal·lació d'actualitzacions de SetelSys en el servidor de producció.
- **ChronoChecker** → Per a la visualització de l'estat del servei de SetelSys.
- **DB Tester** → Validació de la connexió en les diferents bases de dades involucrades en el procés.

*Diagrama arquitectònic de la solució*



*Diagrama arquitectònic del nucli (SetelSysCore)*

En l'apartat [corresponent](#) al desenvolupament, entrarem més en detall en l'objectiu i el funcionament intern de les diferents capes del nucli.

### 3.4 Salvaguarda de les dades i del servidor

Hem trobat oportú dedicar el temps necessari en planificar l'estructura de les còpies de seguretat de les dades i de la configuració inicial del servidor.

#### Backup de les bases de dades de l'ERP

El sistema ERP principal, disposa dels seus corresponents servidors en clúster i d'un sistema de còpies automatitzat, tant pel que fa a les dades d'arxius d'usuari com de les bases de dades. En aquest punt per tant no cal incidir-hi ja que està corresponentment cobert i resolt.

#### Backup de les bases de dades de SetelSys

Pel que fa a les bases de dades generades per l'ús de la nostra aplicació SetelSys, hem planificat el següent sistema:

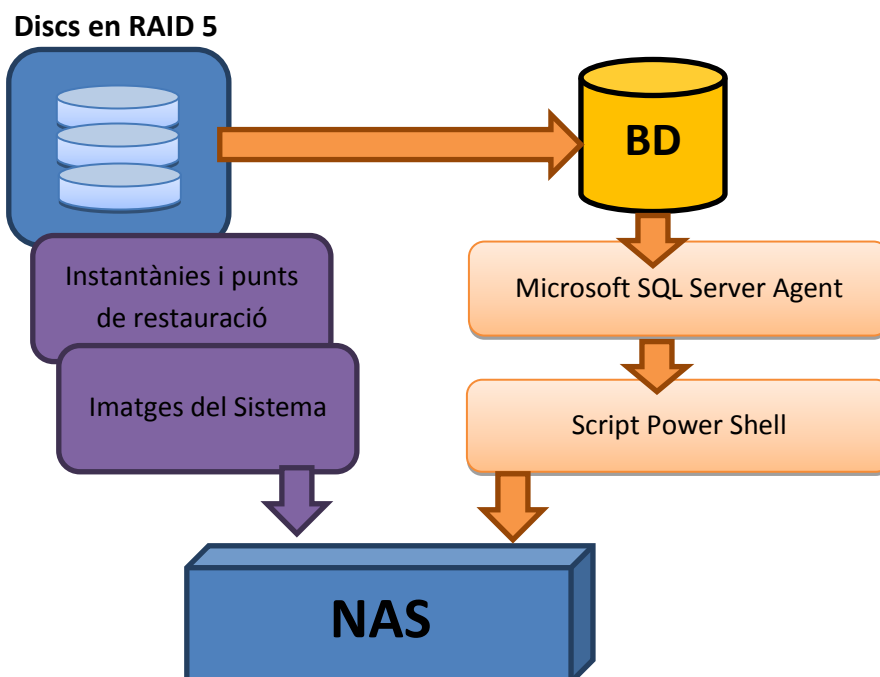
- **Execució de la còpia de seguretat:** Es planifica una tasca utilitzant l'agent del Microsoft SQL Server (SQL Server Agent) que generi una còpia de forma diària a les 12:00h A.M. i la desi en un arxiu localitzat del disc dur amb el nom "BACKUP-dd-mm-yyyy-hh-mm.bak" (dd=dia, mm=mes, yyyy=any, hh=hora, mm=minut).
- **Emmagatzematge de la còpia:** A les 01:00h A.M. s'executa una tasca programada en el servidor (i preparada amb les comandes de Microsoft Power Shell) de forma que guardi l'arxiu generat en el punt anterior en un equip NAS (Network Area Storage). L'entorn de xarxa és Gigabit, amb la qual cosa aquest procés és força ràpid.
- **Neteja de còpies obsoletes:** Aquesta mateixa tasca programada, en el cas que el punt anterior s'executi correctament, s'encarrega d'eliminar còpies velles per evitar un consum innecessari d'espai d'emmagatzematge. Totes les còpies anteriors a 7 dies, són purgades de forma automàtica per aquest script.

### Backup del sistema

Els discos durs del servidor, configurats en un sistema RAID 5, disposen d'un sistema propi de còpia donada l'arquitectura i nivell del sistema RAID. En el cas de l'errada d'algun dels discos, podríem recuperar la resta de les dades amb els existents regenerant el sistema RAID amb els que estarien encara funcionals.

No obstant, per maximitzar la seguretat i salvaguarda de l'equip servidor, hem configurat:

- **Sistema d'instàntànies i punts de restauració** per poder, en qualsevol moment, fer un *rollback* del sistema i tornar al punt anterior.
- Planifiquem de **forma trimestral**, una parada de manteniment que dediquem, entre d'altres punts a revisar, a generar una **imatge exacta del sistema** per poder-la restaurar en cas d'una fallida que no es pogués recuperar per cap de les dos vies anteriors. Tenint en compte que el servidor, a nivell d'aplicacions, no és molt dinàmic (o és sobretot a nivell de bases de dades i aquestes ja queden cobertes segons el protocol establert en l'anterior punt), aquest interval de temps és més que suficient per assegurar la disponibilitat de qualsevol recuperació.



### 3.5 Convenis i nomenclatura utilitzada per a la codificació

De cara a utilitzar una nomenclatura estandaritzada en el procés de desenvolupament, hem marcat un seguit de premises a seguir en compte durant la codificació d'interfícies, classes, propietats i mètodes. Aquesta nomenclatura establerta no només s'ha usat durant les fases de desenvolupament, sinó que s'ha establert també com a nomenclatura per a les fases de proves, especialment durant l'automatització. A més, tal i com s'explica en l'apartat corresponent, també s'ha establert una sèrie de paràmetres dins la nomenclatura per a les funcions de test.

#### 3.5.1 Conveni global

De forma general, aplicarem una normativa global al moment d'escollir el nom d'un control, classe, enumeració, propietats, ... Caldrà complir les següents normes:

- No s'utilitzaran espais
- No s'utilitzaran símbols d'accentuació.
- No s'utilitzaran guions, comes, punts,...
- No s'utilitzaran caràcters com "ç", ni apòstrofs, ...
- Si el nom compren més d'una paraula, la primera lletra s'anotará sempre en majúscula.

Buscarem sempre un nom adequat per identificar-ho, excepte els components globals encarregats de l'arquitectura funcional, els quals utilitzarem noms en clau i breus (Heimdall, Mercury, Seshat, ERPDataCom, Midgard,...)

Exemples: *ConsultarModelsAssignats*, *Cancel·larFinestra*, *NomUsuari*, *LlistaMagatzems*, *ConsultarUnitatsPendants*, *ModificarDocumentVenda*, *EliminarCaixaMagatzem*,...

### 3.5.2 Noms dels controls gràfics de les interfícies d'usuari (UI)

El nom que anotarem als controls estarà format per un prefix i un sufix complint les següents característiques.

**Prefix:** Utilitzarem 3-4 lletres inicials per a definir la classe del control, i ser més fàcil així la seva localització per codi. Exemples:

Sufix	Classes de controls on utilitzar-lo
<b>txt</b>	TextBox, TextBlock
<b>cmd</b>	Button
<b>opt</b>	OptionButton, RadioButton
<b>chk</b>	CheckBox
<b>grid</b>	DataGrid, DataGridView
<b>lbl</b>	Label
<b>cmb</b>	ComboBox
<b>grp</b>	GroupBox
<b>frm</b>	Form
<b>win</b>	ChildWindow
<b>tab</b>	TabCollection
<b>lst</b>	ListBox, ListView
...	...

**Sufix:** Utilitzarem la normativa global.

Aplicant les normes vistes pel prefix i sufix, alguns possibles exemples complets de nom de control podrien ser: *gridAssignarModel*, *cmdCancelar*, *txtNomUsuari*, *lstLlistaMagatzems*, *cmdConsultarUnitatsPendants*,...

### 3.5.3 Nom de les classes

Utilitzarem la normativa global a excepció de les classes representatives dels objectes de bases de dades, en els quals utilitzarem el prefix especial "DB\_" per que quedi sempre constància de que treballem amb objectes que encapsulen les propietats de taules de les bases de dades.

Exemples: *GestorConfiguracioTarifes*, *ParametresPrevisioTraspasos*, *ErrorSistema*,...

### 3.5.4 Nom de les propietats de les classes

En aquest cas utilitzarem dos criteris segons la naturalesa de la propietat:

#### Propietats d'àmbit públic

Utilitzarem la normativa global.

Exemples: *NomUsuari*, *CodiTarifa*, *DataCompra*,...

#### Propietats d'àmbit privat

Sempre les anotarem amb MAJÚSCULA, i utilitzant com a primer caràcter i separador de paraules un guió baix ("\_"). L'objectiu és diferenciar clarament les propietats públiques de les privades.

Exemples: *\_ID\_COMPRA*, *\_NOM\_TARIFA*, *\_CODI\_ARTICLE*

### 3.5.5 Nom dels mètodes i funcions

Utilitzarem la normativa global i posarem especial atenció a denotar una semàntica adequada amb les paraules utilitzades per ajudar a la identificació de la seva funció.

#### Prefixos CRUD

Per anomenar els mètodes o les funcions que es dediquin a l'operativa CRUD (Create – Read – Update – Delete) intentarem sempre utilitzar un mateix prefix.

Taula de prefixos a utilitzar en operatives CRUD:

Operativa	Prefix a utilitzar	Exemples
CREATE	Afegir	AfegirNouUsuariSetelSys(...)
READ	Consultar	ConsultarUsuarisActius()
UPDATE	Modificar	ModificarUsuari(...)
DELETE	Eliminar	EliminarUsuari(...)

#### Atributs dels mètodes i funcions

En els atributs que assignem a la definició d'un mètode o funció, utilitzarem la normativa global. No obstant intentarem definir sempre que sigui possible o interessant (segons criteri del desenvolupador), el sumari de l'element, anotant la descripció de l'ús del mètode o funció així com dels seus paràmetres per permetre una lectura còmoda i intel·ligible en les crides posteriors des d'altres classes.

### 3.5.6 Nom dels delegats, predicats, ...

En el cas d'utilitzar classes meta estructurals com els delegats, predicats i altres, utilitzarem un prefix inicial per a denotar el tipus de representació.

Exemples: *predFiltrarFiles*, *delCarregarInformacio*, ...



### 3.5.7 Resta d'elements de desenvolupament

Per la resta (enumeracions, events, variables, ...) utilitzarem la normativa global.

### 3.5.8 Funcions de test

Les funcions que s'utilitzen a l'hora de codificar els tests, que en el Visual Studio s'identifiquen inicialment amb el *tag* [Test Method], a més a més de complir amb la resta de nomenclatura estàndard que s'han explicat en els apartats anteriors, és important remarcar que s'ha optat per a donar noms molt exemplificadors i identificatius, per a tenir clar quina és la prova que s'està duent a terme en aquella funció.

L'estratègia de proves seleccionada permet que les proves ens si mateixes siguin força *auto-documentades*, en el sentit que són senzilles de seguir i de llegir, i amb la selecció d'un nom clar permet ràpidament entendre les proves a fer.

Alguns dels exemples poden ser:

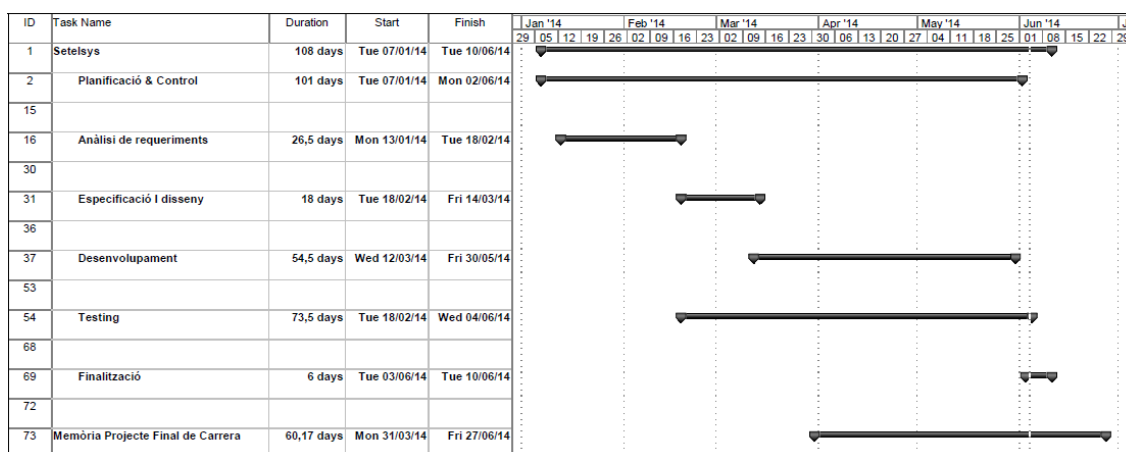
- *CreacióNouUsuariAcces*: provar l'accés a la funcionalitat de creació de nou usuari.
- *CreacióNouUsuariNomObligatori*: prova per a comprovar si la validació de que el nom de l'usuari és obligatori es compleix.

## 4. Planificació

A continuació es presenta la planificació general duta a terme al principi del projecte, i posteriorment s'expandeixen els diferents apartats per a veure en detall les diferents tasques que componen cadascuna de les seccions en les que s'ha dividit el projecte.

### 4.1 Diagrama de Gantt inicial

La versió esquemàtica de la planificació inicial que es va dur a terme, en forma de diagrama de Gantt és la següent:



Tal i com es pot veure, a grans trets s'ha separat per un costat el projecte SetelSys en sí mateix, de la creació de la documentació per al projecte de final de carrera, i dins del projecte SetelSys, s'ha estructurat de la següent forma:

- Planificació i Control
- Anàlisi de Requeriments
- Especificació i Disseny
- Desenvolupament
- Testing
- Finalització

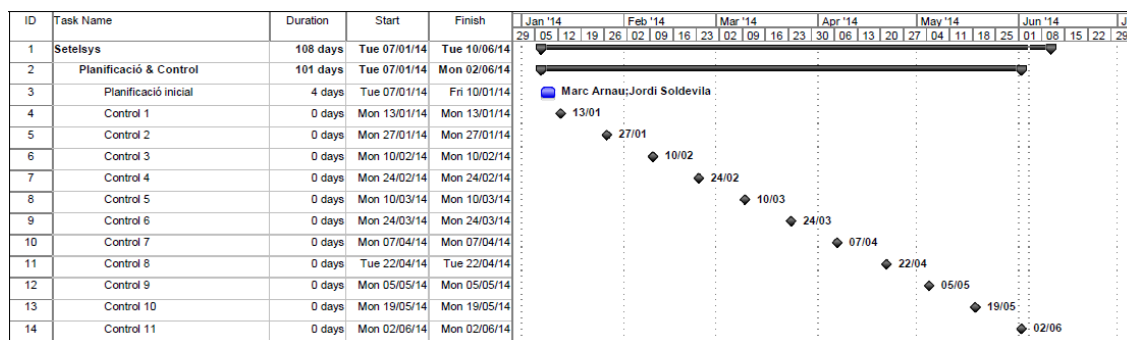
En el següent apartat és detallen cadascun dels apartats, i s'expliquen breument les tasques en que s'han subdividit el projecte, però a nivell general cal remarcar que s'ha optat per treballar de forma conjunta, amb una sèrie de punts de control (tant a nivell de SetelSys com a nivell de la creació de la memòria del projecte) on hem anat compartint els avanços en les diferents tasques compartides, i a la vegada, hem han compartint el detall de les tasques més particulars de cadascú, principalment en les àrees de Desenvolupament i Testing.

També s'ha realitzat un anàlisi econòmic de la solució plantejada, però s'ha optat per a explicar-lo en un [apartat](#) al final de la memòria.

## 4.2 Descripció de les tasques

### 4.2.1 Planificació i Control

El diagrama de Gantt en detall de la fase de **Planificació i Control** és el següent:

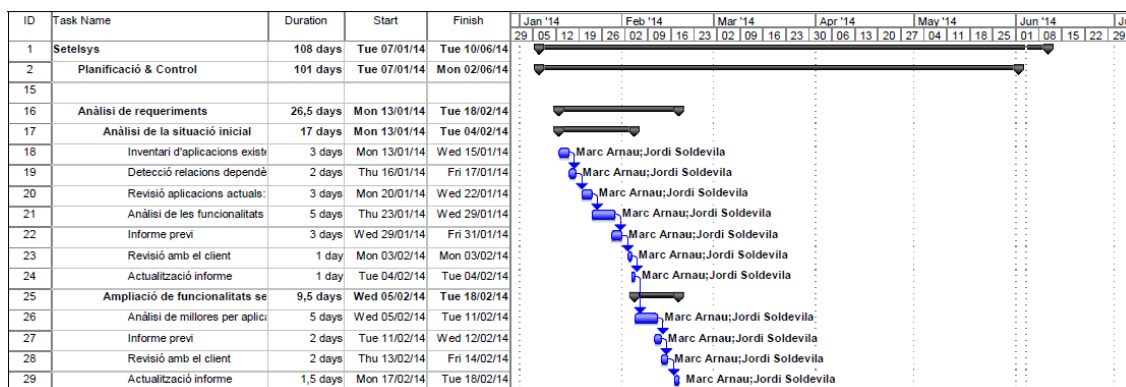


Les seves tasques consisteixen en:

- **Planificació inicial:** tasca integrament dedicada a la planificació de tot el projecte, tenint com a resultats aquests diagrames de Gantt que s'estan analitzant.
- **Punts de control:** al fer un treball en equip, i tot i la interacció força constant, es va optar per a marcar una sèries de fites de control per a revisar planificació, tasques en progrés tant comunes com específiques, modificacions, etc...

## 4.2.2 Anàlisi de Requeriments

El diagrama de Gantt en detall de la fase de **Planificació i Control** és el següent:

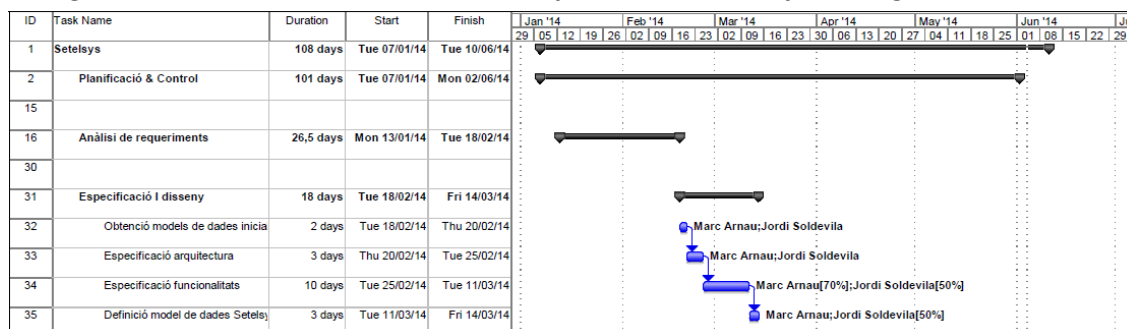


Les seves tasques consisteixen en:

- Anàlisi de la situació inicial
  - **Inventari d'aplicacions existents:** tal i com s'explica en el "Punt de Partida", no es tractava d'un sistema sorgit des de zero, sinó que existia un sistema vigent amb el que s'havia de treballar, i per tant calia inventariar que hi havia.
  - **Detecció relacions dependència entre aplicacions:** un cop inventariades les aplicacions, calia veure quines dependències hi havia entre elles
  - **Revisió aplicacions actuals:** completa i vigència: registrar per a totes les aplicacions existents, l'estat en el que estaven, l'ús que se'n feia així com si calia seguir mantenint-les.
  - **Anàlisi de les funcionalitats actuals:** a part del inventari d'aplicacions, calia veure totes les funcionalitats que es trobaven en el sistema inicial.
  - **Informe previ:** documentació sobre les anteriors tasques d'anàlisi de la situació actual
  - **Revisió amb el client:** revisió del document sorgit de la tasca anterior amb el client per a detectar errors, mancances, millores, etc.
  - **Actualització informe:** processament dels canvis suggerits pel client.
- Ampliació de funcionalitats segons requisits del client
  - **Anàlisi de millores per aplicació:** analitzar totes les millores demanades per part del client.
  - **Informe previ:** documentació sobre les millores a implementar a l'aplicació.
  - **Revisió amb el client:** revisió del document sorgit de la tasca anterior amb el client per a detectar errors, mancances, millores, etc.
  - **Actualització informe:** processament dels canvis suggerits pel client.

### 4.2.3 Especificació i Disseny

El diagrama de Gantt en detall de la fase de **Especificació i Disseny** és el següent:

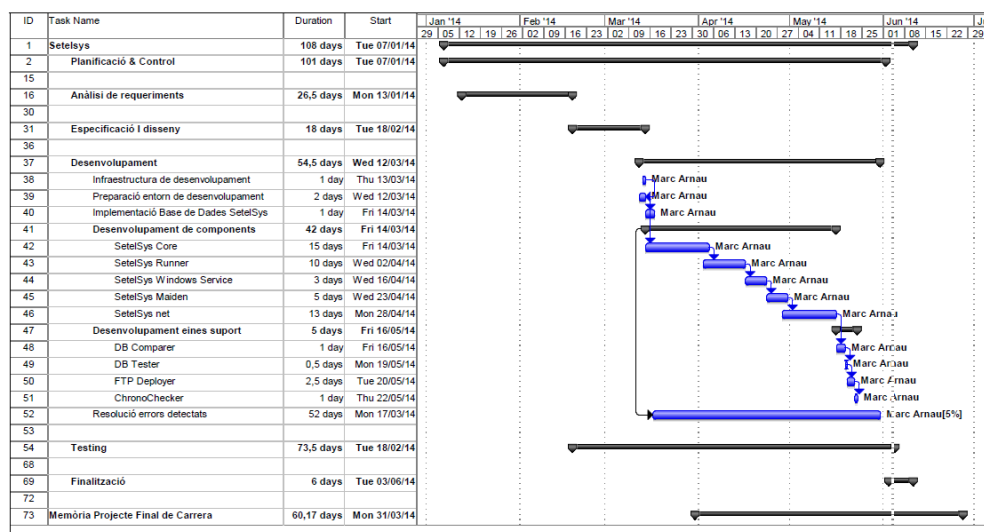


Les seves tasques consisteixen en:

- **Obtenció models de dades inicials:** procés d'enginyeria inversa per a obtenir els models de dades, de les bases de dades inicials en el sistema.
- **Especificació arquitectura:** detall de l'arquitectura de components del sistema, així com les dependències entre els diferents elements.
- **Especificació funcionalitats:** especificació en detall de totes les funcionalitats a implementar a SetelSys.
- **Definició model de dades SetelSys:** especificació del nou model de dades per al nou sistema.

## 4.2.4 Desenvolupament

El diagrama de Gantt en detall de la fase de **Desenvolupament** és el següent:

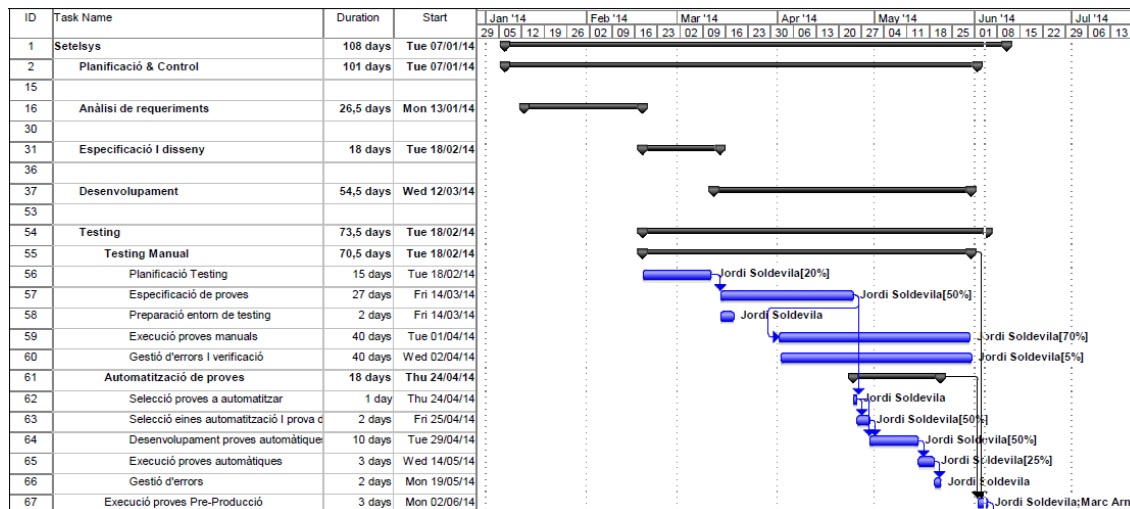


Les seves tasques consisteixen en:

- **Infraestructura de desenvolupament:** instal·lació i validació de les eines i entorn de desenvolupament, necessaris per a dur a terme les tasques posteriors.
- **Preparació entorn de desenvolupament:** posada en marxa del que serà l'entorn de desenvolupament, on s'aniran instal·lant i preparant els diferents elements que s'aniran desenvolupant potseriorment.
- **Implementació Base de Dades SetelSys:** desenvolupar la nova base de dades per al sistema SetelSys segons les especificacions realitzades durant la fase anterior.
- **Desenvolupament de components:** desenvolupament dels diferents components, com són:
  - SetelSys Core
  - SetelSys Runner
  - SetelSys Windows Service
  - SetelSys Maiden
  - SetelSys net
- **Desenvolupament eines suport:** desenvolupament de les eines que suport que s'usaran en el sistam, i que són les següents:
  - DB Comparer
  - DB Tester
  - FTP Deployer
  - ChronoChecker
- **Resolució errors detectats:** correcció dels errors i incidències detectats, tant durant la fase de desenvolupament mateix, com durant tot el procés de proves de software que s'anirà duent a terme.

## 4.2.5 Testing

El diagrama de Gantt en detall de la fase de **Testing i Automatització** és el següent:

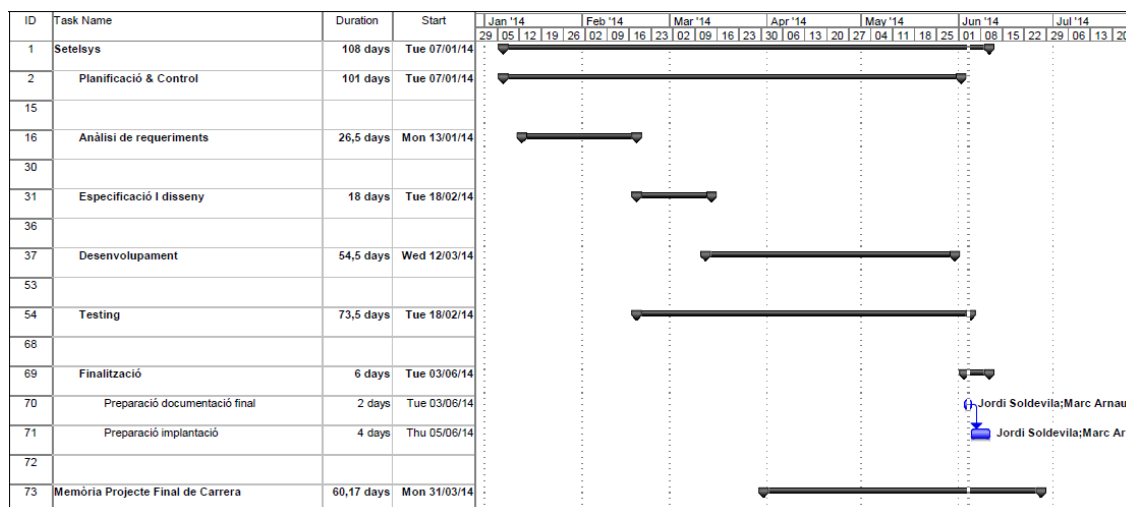


Les seves tasques consisteixen en:

- **Testing Manual**
  - **Planificació Testing:** seguint la metodologia escollida (TMap), fase de planificació, i preparació del pla de proves. Inclou la posada en marxa de l'eina de gestió de proves.
  - **Especificació de proves:** fases de preparació i posterior especificació de les diferents proves a executar durant la posterior fase. Preparació entorn de testing
  - **Execució proves manuals:** execució de les proves prèviament desenvolupades.
  - **Gestió d'errors i verificació:** registre de totes les incidències, errors i millores detectades, i una vegada corregides i entregades, posterior validació, i en alguns casos, re-execució de part de les proves.
- **Automatització de proves**
  - **Selecció proves a automatitzar:** de totes les proves manuals prèviament especificades, realitzar la selecció segons possibilitats i prioritats, de les que seran susceptibles de ser automatitzades.
  - **Selecció eines automatització i prova de concepte:** proves de concepte, per a veure que les eines seleccionades realment funcionen correctament i el projecte d'automatització pot tirar endavant.
  - **Desenvolupament proves automàtiques:** implementació de les proves automatitzades per mitjà de les eines seleccionades.
  - **Execució proves automàtiques:** execució de les proves, per a comprovar-ne la seva validesa, i obtenir els resultats de la qualitat de l'aplicació.
  - **Gestió d'errors:** gestió de qualsevol error detectat, tant a l'aplicació SetelSys, com en les proves automatitzades.
- **Execució proves pre-Producció:** execució de les últimes proves en un entorn de pre-producció per a realitzar les últimes validacions.

## 4.2.6 Finalització

El diagrama de Gantt en detall de la fase de **Finalització** és el següent:

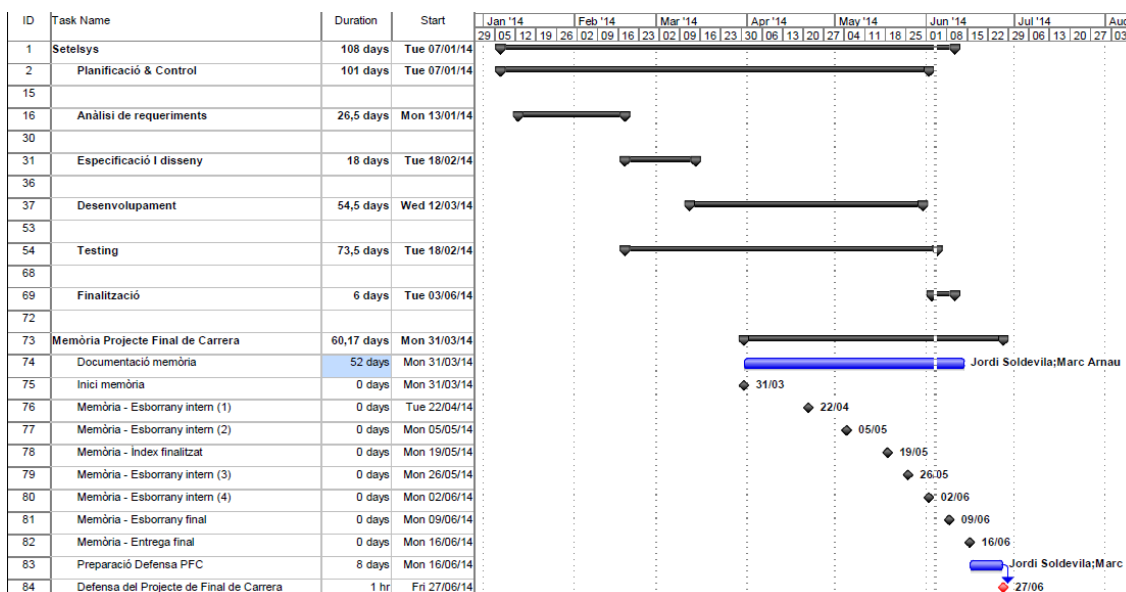


Les seves tasques consisteixen en:

- **Preparació documentació final:** generació de tota la documentació necessària, a tots els àmbits.
- **Preparació implantació:** realització de tots els passos per a dur a terme la implementació.

## 4.2.7 Memòria del Projecte de Final de Carrera

A més a més del propi projecte, ja vam planificar des del inici com duríem a terme la creació de la documentació per al Projecte de Final de Carrera. La planificació plantejada inicialment era:





### 4.3 Projecte

Per a l'enregistrament de les hores destinades en les diferents etapes del projecte, utilitzarem una eina disponible de forma gratuïta a la xarxa que ens permetrà utilitzar un espai de treball conjunt entre nosaltres on assignarem diferents projectes associats a les diferents etapes del nostre projecte global.

D'aquesta manera podrem comptabilitzar de forma còmoda i compartida el temps real destinat a l'elaboració de les diferents etapes i poder contrastar possibles desviacions finals en la finalització del projecte.

L'eina en qüestió que utilitzarem és **"Toggl"**<sup>12</sup>. És una solució web on-line que permet:

- Crear espais de treball compartits entre usuaris.
- Crear clients
- Crear projectes
- Associar projectes a clients
- Anotar tasques dedicades a un projecte, anotant-ne el temps destinat i dia, i una descripció de la tasca.
- Utilitzar el seu propi temporitzador per iniciar, continuar o parar el càlcul de temps.
- Generar informes mensuals, setmanals, entre dates, etc, on apareix un resum dels temps dedicat a cada projecte i podent visualitzar el detall de les tasques dutes a terme.

També disposen d'una versió disponible per Escriptori i per terminals mòbils (Android i iOS).

---

<sup>12</sup> Pàgina web Toggl: [www.toggl.com](http://www.toggl.com)

## 5. Anàlisi de requeriments


Per a poder desenvolupar un correcte anàlisi de requeriments, i veient la situació actual del sistema (aplicacions obsoletes, redundants, desconeixement d'ús actiu,...) haurem de seguir les següents etapes per poder concloure un anàlisi adequat a la realitat i a les necessitats actuals de l'empresa i el seu sistema satèl·lit actual:

1. Inventariar totes les aplicacions existents.
2. Definir les relacions existents entre les diferents aplicacions.
3. Determinar la completessa i vigència de les diferents aplicacions. Cal determinar quines són completes funcionalment i vigents pel que fa a la utilització actual en el sistema.
4. Analitzar la funcionalitat actual de les aplicacions, ja que en molts casos els propis usuaris desconeixen el patró de funcionament i les variables que es tenen en compte en cada aplicació
5. Contrastar les funcionalitats actuals amb el client.
6. Complementar les funcionalitats actuals amb les noves desitjades pel client.
7. Elaborar un primer informe amb el contingut dels diferents punts i sotmetre'l a revisió amb el client fins determinar, amb la màxima fidelitat possible, la llista de requeriments finals del sistema SetelSys.



### 5.1 Inventari d'aplicacions satèl·lit

Cal en primer lloc elaborar un inventari de les aplicacions existents. Anotem a continuació el resultat de l'inventari agrupat pels blocs que ja hem presentat anteriorment.

Per cada aplicació de Microsoft Access present, anotem:

- Nom descriptiu (S'anota en castellà per ser el nom en que es coneix dins l'empresa i evitar confusions a l'hora d'identificar l'aplicació)
- Breu descripció textual
- Ruta d'accés en xarxa a l'arxiu
- En el cas de que l'aplicació s'executi com un procés automatitzat, ho indiquem amb la icona 

## 5.1.1 Bloc 1: Logística


Nom	Breu descripció
<b>Datos</b>	<b>Actualitza les factures</b> de comandes ja enviades amb la informació referent al número d'expedició de l'albarà associat.
<b>Expedición Seur</b>	<a href="\\proliant\c\datos seur\Datos expedicion seur.mdb">\\proliant\c\datos seur\Datos expedicion seur.mdb</a>
 <b>Pedidos</b>	<b>Genera</b> els traspasos entre les diferents botigues i també el magatzem central, per fer-los-hi arribar el material necessari per poder servir les seves comandes de clients que estan pendents de rebre material <b>en estoc</b> . <a href="\\fileserv\...\Pedidos_tiendas\pedidos.mdb">\\fileserv\...\Pedidos_tiendas\pedidos.mdb</a>
<b>Actualizar Pendientes Servir</b>	Actualitza les dades dels aplicatius utilitzats per a controlar les comandes pendents de servir al client final. Així es pot disposar de la informació actualitzada pel que fa a la disponibilitat dels articles a servir al client, en aplicatius posteriors. <a href="\\fileserv\...\pendientes_servir_y3\Actualizar pdtes_servir.mdb">\\fileserv\...\pendientes_servir_y3\Actualizar pdtes_servir.mdb</a>
 <b>Borrar Etiquetas</b>	<b>Esborra</b> uns documents que es generen en el sistema ERP diàriament per la impressió d'etiquetes d'articles utilitzant una sèrie de documents temporal. <a href="\\Server\C\informes josema\borrar_etiquetas.mdb">\\Server\C\informes josema\borrar_etiquetas.mdb</a>
<b>Pedidos Tiendas</b>	Llistats per a poder llistar el material que rebran les diferents botigues, des de altres botigues o el magatzem central, i així poder saber a qui va destinat cada article que els hi arribi amb la remesa del transportista. <a href="\\fileserv\...\Pedidos_tiendas\x.mdb">\\fileserv\...\Pedidos_tiendas\x.mdb</a> (donde x = nombre de la tienda)
<b>Pendientes Recibir Front</b>	Permet llistar els traspasos pendents d'acceptar dins el programari "Front" (TPV), des de les diferents botigues. És important acceptar-los, ja que sinó aquests articles del traspàs, no entrarien dins l'estoc del seu magatzem. <a href="\\Fileserv\userdata\Access_database\pdtes_recibir_front\pdtes.mdb">\\Fileserv\userdata\Access_database\pdtes_recibir_front\pdtes.mdb</a>
<b>Pendientes de Servir Tiendas</b>	Gestió de les comandes pendents de servir al client final des de les botigues (recollida a la botiga). Des d'aquest aplicatius poden visualitzar l'estat de les comandes pendents, així com anotar comentaris i notes en el seguiment de la comanda (disponibilitat, contacte amb el client, ...). <a href="\\Fileserv\userdata\Access_database\pendientes_servir_y3\pendientes.mdb">\\Fileserv\userdata\Access_database\pendientes_servir_y3\pendientes.mdb</a>
<b>Pendientes Servir (Central / Web)</b>	Igual que l'anterior aplicació, però en aquest cas gestiona les comandes pendents de servir directament al client des del magatzem central (enviament a domicili). <a href="\\Fileserv\userdata\Access_database\pendientes_servir\pendientes.mdb">\\Fileserv\userdata\Access_database\pendientes_servir\pendientes.mdb</a>
<b>Servir Todo Tiendas</b>	Utilitzat per a servir el material de comandes pendents que s'envien directament al client final. L'aplicatiu té dos funcionalitats bàsiques: <ol style="list-style-type: none"> <li>1. Moure el material des de les botigues cap al magatzem central, si no hi ha estoc disponible en el central.</li> <li>2. Una vegada rebut el material, generar els albarans i preparar el material per enviar-lo al client final.</li> </ol> <a href="\\Fileserv\userdata\Access_database\servir_desde_tiendas\servir_TODO_tiendas.mdb">\\Fileserv\userdata\Access_database\servir_desde_tiendas\servir_TODO_tiendas.mdb</a>
<b>Quitar de traspasos</b>	Extreu dels traspasos d'articles des de la central cap a les botigues, que es preparen durant el cap de setmana, articles que poden haver passat a ser necessaris per alguna comanda pendent d'un client o botiga durant l'impàs de temps del cap de setmana. <a href="\\Fileserv\userdata\Access_database\pedidos-traspasos\traspasos.mdb">\\Fileserv\userdata\Access_database\pedidos-traspasos\traspasos.mdb</a>

<b>Caja Almacén</b>	Llistat de caixes utilitzat per a organitzar el material a preparar pels enviaments a domicili dels clients finals. <a href="\\\\Fileserver\\userdata\\Access_database\\cajas_almacen\\cajas_almacen.mdb">\\\\Fileserver\\userdata\\Access_database\\cajas_almacen\\cajas_almacen.mdb</a>
<b>Etiquetas Almacén</b>	Genera un llistat d'etiquetes per a les caixes, una vegada hem especificat certs valors d'entrada (botiga, sèrie, nº inicial i nº final). <a href="\\\\Fileserver\\userdata\\Access_database\\EtiquetasAlmacen\\Etiquetas_almacen.mdb">\\\\Fileserver\\userdata\\Access_database\\EtiquetasAlmacen\\Etiquetas_almacen.mdb</a>

### 5.1.2 Bloc 2: Articles, estocs i inventaris

Nom	Breu Descripció
<b>Artículos</b>	Permet crear articles directament en la base de dades del ERP, aprofitant la creació per poder generar les talles i colors de l'article, així com els preus en diferents tarifes i estocs mínims. <a href="\\\\Fileserver\\userdata\\Access_database\\articulos\\articulos.mdb">\\\\Fileserver\\userdata\\Access_database\\articulos\\articulos.mdb</a>
<b>Mínimos</b>	Aplicació per a actualitzar els valors mínims d'estoc desitjats per diferents articles i diferents magatzems especificats en un full de càlcul d'entrada en format <b>Microsoft Excel</b> . <a href="\\\\proliant\\c\\informes_josema\\minimos.mdb">\\\\proliant\\c\\informes_josema\\minimos.mdb</a>
<b>Regularizaciones</b>	Permet llistar, segons diferents criteris i paràmetres, regularitzacions efectuades en diferents botigues o a nivell global, per unes dates concretes o per mesos. <a href="\\\\Fileserver\\userdata\\Access_database\\regularizaciones\\regularizaciones.mdb">\\\\Fileserver\\userdata\\Access_database\\regularizaciones\\regularizaciones.mdb</a>
<b>Artículos en Tienda</b>	Aplicació utilitzada en els botigues per a la visualització dels estocs de certs articles en la seva pròpia botiga o en les altres botigues de l'empresa. <a href="\\\\Fileserver\\userdata\\Access_database\\stock_tienda\\articulos_en_tienda">\\\\Fileserver\\userdata\\Access_database\\stock_tienda\\articulos_en_tienda</a>
<b>Poner Ubicaciones</b>	<b>Actualitza</b> la ubicació dels articles del magatzem central, a partir d'un full de càlcul en format <b>Microsoft Excel</b> . Per fer-ho es basa en la lectura efectuada amb un terminal de ma, de les diferents ubicacions del magatzem i els articles que hi ha guardats, llegint el codi de barres identificador, tant de les ubicacions com els articles. <a href="\\\\proliant\\c\\informes_josema\\poner_ubicaciones.mdb">\\\\proliant\\c\\informes_josema\\poner_ubicaciones.mdb</a>
<b>Actualizar Peso</b>	Permet modificar els diferents articles de la base de dades del ERP, perquè actualitzi el camp associat al pes de l'article. Per fer-ho es basa en un barem emmagatzemat en una taula local, on associa diferents pesos a diferents famílies, subfamílies, departaments i seccions. <a href="\\\\proliant\\c\\pesos\\actualizar_peso.mdb">\\\\proliant\\c\\pesos\\actualizar_peso.mdb</a>

### 5.1.3 Bloc 3: Tarifes i preus

Nom	Breu Descripció
<b>Actualizar Tarifas España</b>	<p>Procés automàtic que s'encarrega de calcular i regenerar certes tarifes de preus associades als articles. Partint d'unes tarifes bàsiques o "mare", regenera la resta de tarifes aplicant els criteris que pertoca a cada tarifa.</p> <p><a href="\\Server\C\informes josema\actualizar tarifas esp.mdb">\\Server\C\informes josema\actualizar tarifas esp.mdb</a></p>
<b>Recuperar PVP</b> 	<p>Guarda una còpia de una de les tarifes bàsiques del sistema ERP principal, en una taula local de la pròpia aplicació. En cas d'algun error humà a l'eliminar la tarifa, sempre es pot recuperar més ràpidament que no pas accedint a les còpies de seguretat de les bases de dades SQL Server emmagatzemades en una cinta magnètica.</p> <p><a href="\\Server2\e\josema\recuperar_pvp.mdb">\\Server2\e\josema\recuperar_pvp.mdb</a></p>
<b>Importación Oferta Andorra</b>	<p>Utilitza la web com a sistema d'intercanvi de dades o emmagatzematge intermig per tal de importar els preus de la tarifa d'oferta de la base de dades de la seu d'Andorra dins la base de dades de la seu d'Espanya.</p> <p><a href="\\Server2\e\josema\importacion oferta andorra.mdb">\\Server2\e\josema\importacion oferta andorra.mdb</a></p>
<b>Exportación PVP España</b>	<p>Utilitza, com en el cas anterior, la base de dades de la web per exportar els preus de la tarifa PVP de la base de dades d'Espanya, per una posterior importació en la base de dades d'Andorra.</p> <p><a href="\\Server2\e\josema\exportacion pvp espanya.mdb">\\Server2\e\josema\exportacion pvp espanya.mdb</a></p>

### 5.1.4 Bloc 4: CRM, consultes i altres

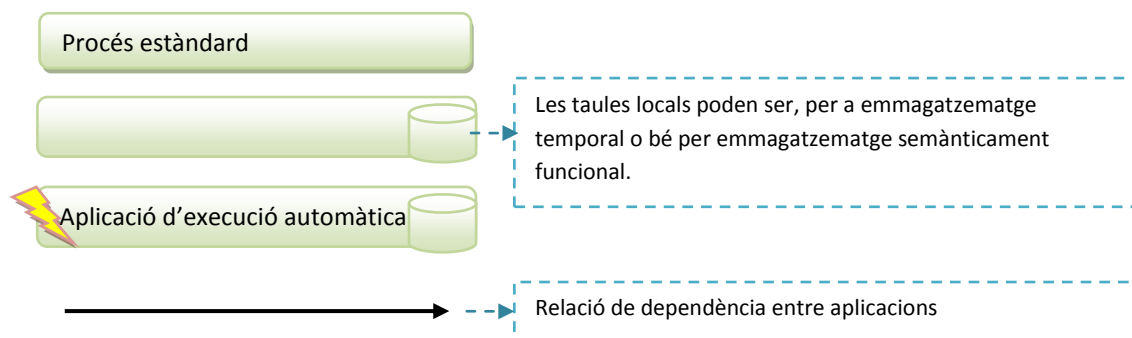
Nom	Breu Descripció
<b>Actualizar Bonos</b>	<p>Actualitza l'estat de les targetes regal perquè estiguin actualitzades tant en la base de dades de l'ERP com en la de la web per que no es puguin consumir més d'una vegada en les compres.</p> <p><a href="\\Server2\c\targetas regalo\Actualizar bonos.mdb">\\Server2\c\targetas regalo\Actualizar bonos.mdb</a></p>
<b>Motos Clientes</b>	<p>Permet assignar a un client, el model de moto del que disposa per, posteriorment, poder personalitzar l'oferta d'articles que s'ajustin a la seva moto. El client ens confirma ja en el moment de l'inventari, que aquesta aplicació estava en procés de creació i actualment es vol deixar de banda per fer-lo més acuradament en un futur.</p> <p><a href="\\Fileserver\userdata\Access database\motos clientes\motos clientes.mdb">\\Fileserver\userdata\Access database\motos clientes\motos clientes.mdb</a></p>
<b>Tarjetas Fidelización</b>	<p>Utilitzat per a actualitzar les dades de les targetes de fidelització de clients. Associa els nous clients creats a una targeta acumulativa de punts per compres utilitzant el codi intern de client del ERP.</p> <p><a href="\\Fileserver\userdata\Access database\fidelizacion\tarjetas fidelizacions.mdb">\\Fileserver\userdata\Access database\fidelizacion\tarjetas fidelizacions.mdb</a></p>
<b>Explotación</b>	<p>Permet generar llistats de les explotacions comptables de les diferents botigues, desglossant-les en base d'una prèvia parametrització.</p> <p><a href="\\Proliant\c\explotacion11\explotacion.mdb">\\Proliant\c\explotacion11\explotacion.mdb</a></p>
<b>Anticipos Clientes</b>	<p>Aplicació utilitzada per a actualitzar els noms dels clients en els pagaments avançats dels clients en els assentaments comptables.</p> <p><a href="\\Server2\ANTICIPOS CLIENTES.mdb">\\Server2\ANTICIPOS CLIENTES.mdb</a></p>

## 5.2 Relacions de dependència entre els diferents blocs d'aplicacions

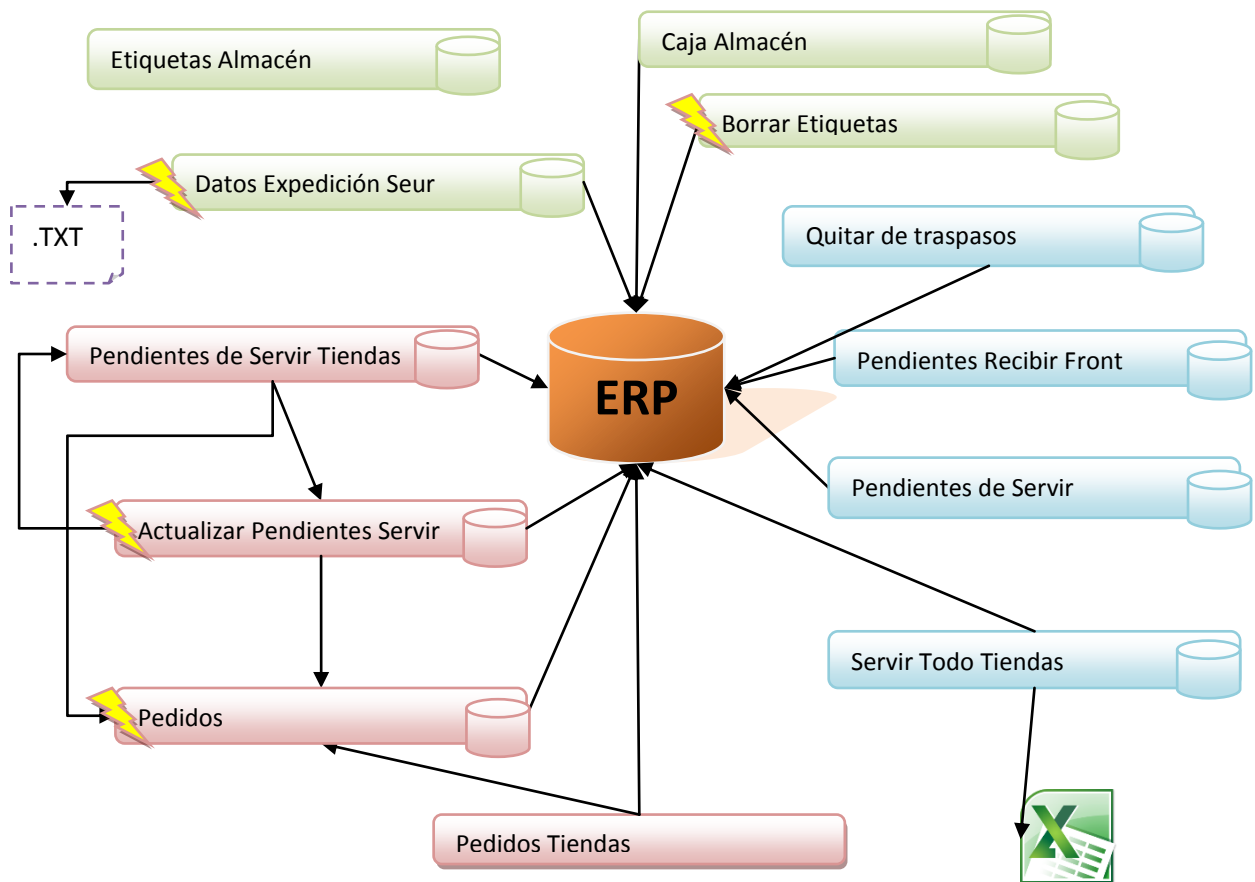
Una vegada inventariades totes les aplicacions del sistema satèl·lit, cal identificar les dependències actuals entre les aplicacions de diferents blocs o d'un mateix bloc. Com hem apuntat anteriorment, algunes de les taules d'aquestes aplicacions de *Microsoft Access* contenen enllaços no només cap a la base de dades principal, sinó a altres de les aplicacions satèl·lit.

En aquest apartat doncs, anotem mitjançant relacions gràfiques, les diferents dependències entre els diferents blocs d'aplicacions.

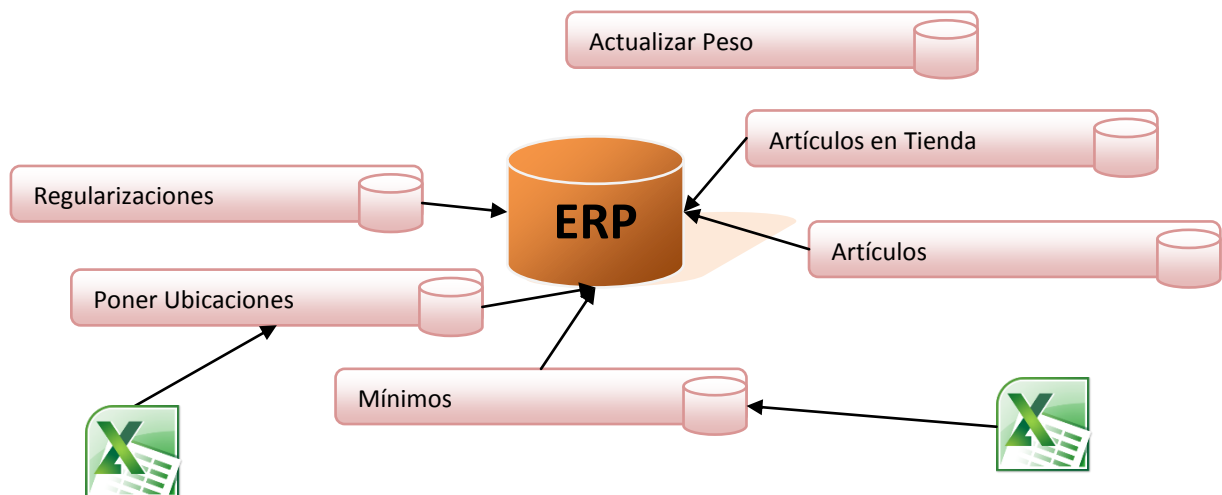
Per a poder interpretar correctament els gràfics de dependències, adjuntem a continuació una llegenda descriptiva dels diferents elements que intervenen en ells:



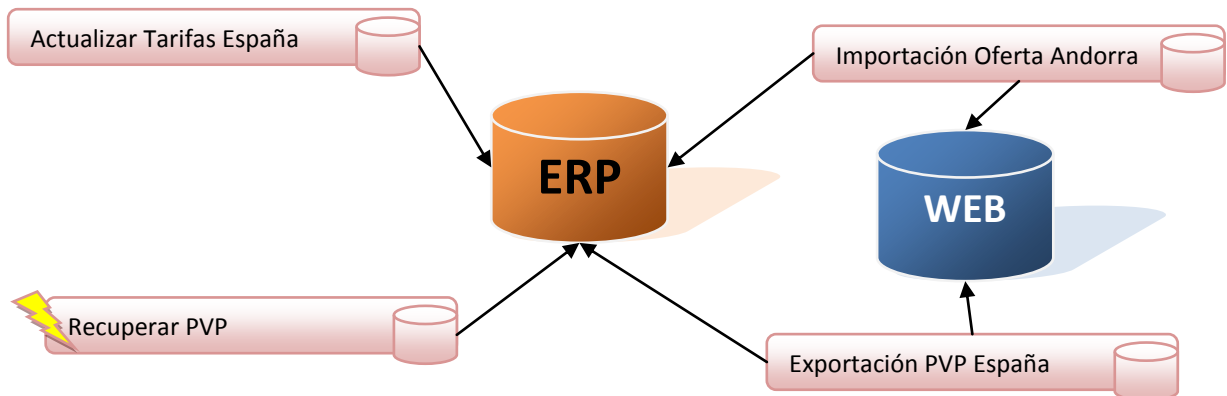
## 5.2.1 Bloc 1: Logística



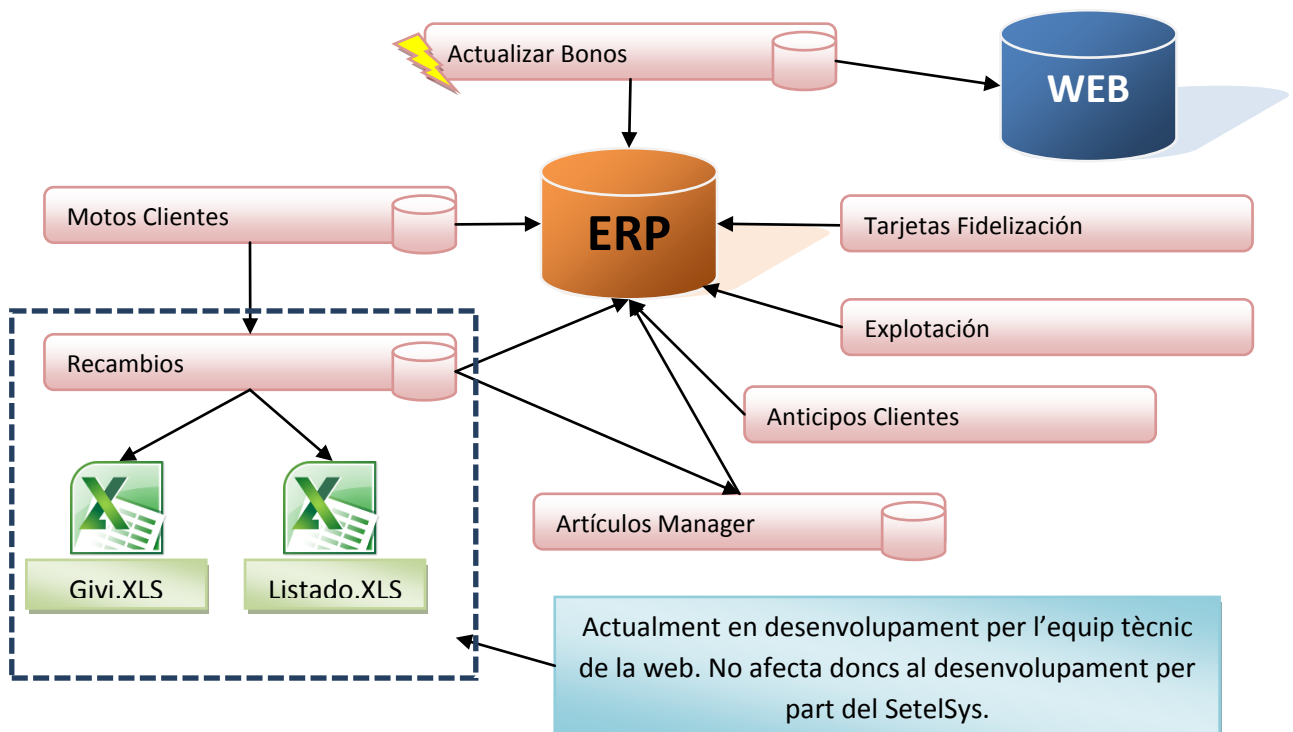
## 5.2.2 Bloc 2: Articles, estocs i inventaris



### 5.2.3 Bloc 3: Tarifes i preus



### 5.2.4 Bloc 4: CRM, consultes i altres





### 5.2.5 Conclusions sobre les dependències entre aplicacions


Com s'ha pogut veure en els gràfics dels apartats anteriors, el nivell de dependència de les aplicacions entre si, és exageradament elevat. Com hem anotat en anteriors punts, això implica:

- Efectes col·laterals no desitjats a l'efectuar canvis sobre una aplicació amb dependències associades.
- Si alguna aplicació en la cadena de dependències pateix algun error d'execució i les seves dades no resulten actualitzades correctament o incoherents, pot afectar la resta d'aplicacions amb les que existeix algun tipus de dependència.
- Es fa molt difícil establir un mecanisme de control en un marc de treball tant fragmentat, disseminat i sense mecanismes de control d'errors nadius.
- La distribució de totes les aplicacions en diferents plataformes hardware, fa que en cas de caiguda d'algun dels servidors, la resta d'aplicacions quedin inconnexes i indueixin a errades o incoherències de dades.
- Excessiva dependència entre aplicatius que fa difícil l'escalabilitat i modificabilitat del sistema, a part d'existir diferents punts on es pot trencar la coherència de les dades i els resultats.



### 5.3 Completesa i vigència de les aplicacions

Després d'haver recollit i catalogat les diferents aplicacions existents al sistema actual, i haver-ne estudiat la funcionalitat i l'objectiu, cal determinar amb el client quines d'aquestes aplicacions són completes pel que fa a la seva funcionalitat i vigents per la seva utilització.

En aquest punt ens reunim amb el client i analitzem el resultat del inventari per aconseguir fer un primer filtrat d'aplicacions on cal focalitzar l'anàlisi i especificació. Citarem a continuació els diferents blocs, detallant-ne les aplicacions que els componen tal i com hem vist en l'apartat 5.1 (*Inventari d'aplicacions*) i anotant si aquestes són completes funcionalment i vigents.

*Nota: Recordem que, com en l'anterior apartat, per anotar que l'aplicatiu és un procés automatitzat, utilitzarem la icona:* 

#### 5.3.1 Bloc 1: Logística

Nom		Completa	Vigent
Datos Expedición Seur		✓	✓
Pedidos 		✗	✓
Actualizar Pendientes Servir		✗	✓
Borrar Etiquetas 		✓	✓
Pedidos Tiendas		✗	✓
Pendientes Recibir Front		✓	✓
Pendientes de Servir Tiendas		✗	✓
Pendientes de Servir (Central/Web)		✗	✓
Servir Todo Tiendas		✗	✓
Quitar de traspasos		✓	✓
Caja Almacén		✗	✓
Etiquetas Almacén		✓	✓

### 5.3.2 Bloc 2: Articles, estocs i inventaris

Nom	Completa	Vigent
Artículos	✓	✓
Mínimos	✗	✓
Regularizaciones	⊘	✗
Artículos en Tienda	⊘	✗
Poner Ubicaciones	✗	✓
Actualizar Peso	✗	✓

### 5.3.3 Bloc 3: Tarifas i preus

Nom	Completa	Vigent
Actualizar Tarifas España	✗	✓
Recuperar PVP ⚡	✗	✓
Importación Oferta Andorra	⊘	✗
Exportación PVP España	⊘	✗

### 5.3.4 Bloc 4: CRM, consultes i altres

Nom	Completa	Vigent
Actualizar Bonos	✓	✓
Motos Clientes	⊘	✗
Tarjetas Fidelización	✓	✓
Explotación	⊘	✗
Anticipos Clientes	✓	✓

### 5.3.5 Conclusions sobre la completeness i vigència de les aplicacions actuals

Com hem pogut veure en els apartats anteriors, el sistema actual compren moltes aplicacions que tenen necessitats funcionals a satisfer, i algunes d'elles que ja no tenen cabuda en el sistema al no ser utilitzades.

Els motius pels quals algunes d'aquestes aplicacions han quedat obsoletes són:

- El sistema ERP Principal ha donat solució, en versions més actuals, a la capacitat de gestionar des del seu motor, algunes de les necessitats.
- Algunes aplicacions del sistema satèl·lit han absorbit i implementat part de les funcionalitats necessàries d'altres aplicacions.
- Canvis en la lògica de processos de l'empresa, que ha anul·lat la necessitat de certes aplicacions.

D'altra banda tota les noves necessitats detectades en les diferents aplicacions i que fa no considerar-les completes provenen dels requeriments de:

- Adaptar el sistema per donar resposta a les necessitats creixents de la lògica de processos de l'empresa. Cal donar una resposta al present i futur de l'empresa.
- Millorar la qualitat de les aplicacions en termes d'agilitat, facilitat, rapidesa i autonomia.

## 5.4 Funcionalitat actual de les aplicacions vigents

En molts casos, les aplicacions actuals del sistema satèl·lit efectuen unes consideracions, parametritzacions i criteris de decisió, de les quals el propi client desconeix el detall exacte. En part degut per una documentació inexistent dels processos i d'altra banda per tractar-se d'aplicacions molt vives i dinàmiques a nivell d'adaptació a l'entorn de l'empresa.

Per això és indispensable determinar el funcionament exacte de les aplicacions actuals per a poder-ho contrastar en el client tal i com veurem en el següent punt.

De cara a fer la memòria més amena i lleugera en la lectura, només introduïrem aquí l'anàlisi de les aplicacions que considerem més important i interessants de les existents, i que requereixen modificacions i/o ampliacions en les funcionalitats.

### 5.4.1 Aplicació "Pedidos" (Bloc 1: Logística)

#### Objecte de procés

Les comandes que ha de tractar aquest procés automatitzat, són totes aquelles les quals el número de sèrie de la comanda del client final compleixi la norma:

#### **E(1X)P**

On **X** és el número identificador de la sèrie de documents d'una botiga: E11P, E12P, E12P...

#### Descripció de l'aplicació

Procés automatitzat que juntament amb els aplicatius següents conforma el nucli neuràlgic del bloc logístic i del control de comandes pendents:

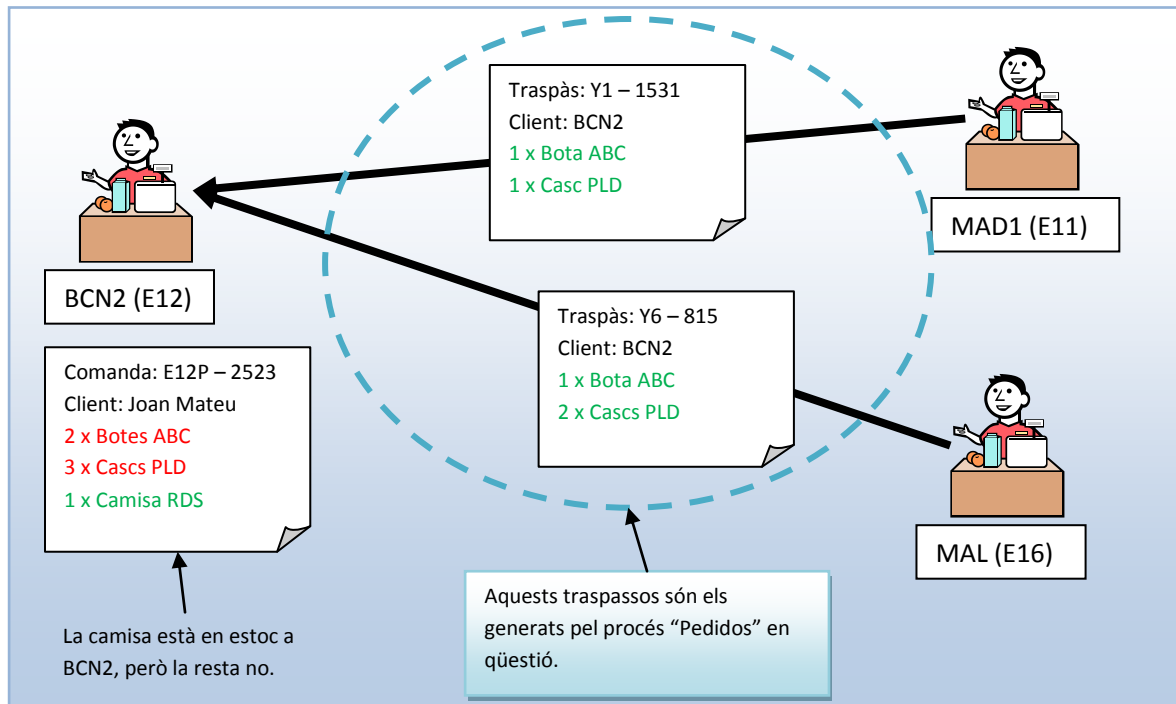
- Servir todo tiendas
- Actualizar pendientes de servir
- Pendientes de servir
- Pendientes de servir tiendas

El procés s'encarrega de generar els traspassos entre botigues per a subministrar els articles pendents per a comandes de clients que recolliran el material a les botigues. Recull totes les comandes pendents de servir al client, de la última setmana (últims 7 dies), i cerca si existeix alguna botiga amb existències d'aquests articles.

Si es el cas, genera un traspàs entre les botigues (sèrie documental Y(x) on x = botiga des d'on es traspassa el material). Aquest traspàs l'accepta la botiga que rep el material a través de l'aplicatiu de TPV i així queda en estoc al seu magatzem.

Posem un exemple per a visualitzar millor el seu funcionament. Cal considerar per aquest exemple:

- BCN2, MAD1, MAL: Són en aquest cas algunes de les botigues existents a l'empresa situades en diferents localitzacions geogràfiques (Barcelona, Madrid, Màlaga).
- Els traspassos són albarans de compra-venta dins l'empresa que es generen en l'ERP utilitzant una sèrie i numeració concreta per aquesta tipologia de documents.



Analitzem una mica el contingut del gràfic:

- En Joan Mateu visita la botiga de BCN2 i fa la seva comanda. Només un dels articles està en estoc, la resta queda pendent d'entregar al client.
- El dijous pel matí, a l'executar-se el procés "PEDIDOS":
  1. Es cerquen botigues on hi hagi estoc del material pendent per la comanda del Joan. Si en una botiga concreta existeix disponibilitat de tot el material, s'envia tot aquest material des d'una sola botiga. En cas contrari s'envia des de diferents botigues per tal de poder reservar el màxim de material per la comanda. En el cas del Joan, una part surt de Madrid i una altra de Màlaga.
  2. Es generen uns traspassos entre les botigues (un amb sèrie Y6, i un amb Y1) amb cost 0, per no desvirtuar les estadístiques de ventes ni el cost de l'article.
  3. Durant el matí del dijous, les botigues reben un llistat agrupat de tot el material que han d'enviar a altres botigues per satisfer comandes pendents. En el cas de Madrid, haurà de preparar "1 x Bota ABC" + "1 Casc PLD", a part de la resta de material necessari per altres comandes acumulades (aquí només analitzem el cas del Joan Mateu).

4. Les botigues s'envien el material entre elles durant la última hora del Dijous al matí.
5. El divendres, les altres botigues reben el material i accepten el traspàs rebut des del TPV del ERP, deixant en estoc tot el material. Al rebre el material, preparen el material i truquen al client, si és el cas. En aquest punt disposen d'un llistat de les comandes que es poden entregar completament i les que només es poden parcialment, per poder avisar al client sobre l'estat de la seva comanda. En el cas del Joan Mateu, ha rebut tot el material i se'l trucarà per comunicar-li.

### **Mancança d'estoc global**

En el cas de que no existeixi estoc en cap magatzem de les botigues existents en l'empresa (incloent el magatzem central), l'aplicatiu procedeix de la següent forma:

- Genera una comanda amb sèrie Y3 i magatzem E13 (central), amb el client associat a la botiga on està pendent la comanda.

Des del magatzem, es llista el material pendent de servir a les botigues i, si és el cas, es procedeix a fer les comandes de compra als proveïdors dins el ERP.

Una vegada arriba el material pendent des del proveïdor, s'envia el material a les botigues perquè puguin servir-lo al client.

### **Prioritats d'enviament entre botigues**

Existeix una prioritat d'assignació entre els magatzems quan existeix estoc en més d'un d'ells. Aquestes prioritats s'utilitzen per diferents criteris:

- a) Magatzems secundaris: Sovint interessa treure material primerament de magatzems més secundaris i que per tant no requereixen de tant material en estoc. En aquest cas es posaran pròxims en la cadena de prioritats.
- b) Proximitat geogràfica: Per qüestions de costos de transport es vol minimitzar el cost del moviment de material entre botigues i per tant les botigues més pròximes geogràficament requeriran estar al inici de la cadena de prioritats.

En tots els casos, el magatzem més prioritari des d'on s'ha d'intentar fer sortir el material és el magatzem central, per evitar treure estocs de les botigues, i les menys prioritàries són les botigues d'Andorra, per evitar costos aranzelaris.

Aquesta assignació de prioritats, actualment no és modificable d'una forma fàcil i àgil. Implica moltes modificacions de codi i estructura del mateix.

## Notificació dels llistats

Al finalitzar tot el procés, l'aplicació envia un missatge a través de correu electrònic a cada una de les botigues amb un informe que conté tota la llista de clients ordenats alfabèticament, els quals tenen reserves fetes a la botiga durant les dates compreses pel tractament de l'aplicació. Així tenen un resum del material que els hi arriba de les diferents botigues.

Els camps que conté aquest informe son:

Nom Client	Sèrie	Número	Telèfon Client	Articles			Magatzem
				Referència	Color	Talla	

## Llistats de control

Associada a aquesta aplicació, es requereix d'uns llistats de control per a cada una de les botigues per a permetre un correcte seguiment de les comandes:

1. Llistat de material reservat, pendent de recollir pel client final. Els camps que haurà de contenir són:

Client	Comanda (Núm.)	Article				Informació de l'estoc				Data Última Compra
		Ref.	Color	Talla	Desc.	Unit. Reservades	Unit. Estoc	Estoc Total	Quantitat Reserves	

2. Llistat del material pendent de rebre des de la central (de la sèrie Y3)



## Faltes d'estoc

En certs casos es poden produir desajustaments entre l'estoc que indica el sistema i l'estoc realment existent o disponible a la botiga (algunes vegades errors inherents a l'ERP, en d'altres errors d'origen humà).

En aquest cas utilitzen una tècnica una mica precària, basada en insereix una referència especialitzada per aquests casos anomenada "DOBLAR". Això fa que el mateix sistema actual detecti aquestes línies i les interpreti com que l'article de la línia anterior, tot i que en el sistema ERP disposa d'estoc, des de la botiga han detectat que realment aquest estoc no existeix.

Si no utilitzen aquest mecanisme, el sistema actual creu que realment ja disposa d'aquest article a la pròpia botiga i per tant no li fa arribar el material d'altres on també n'hi hagi.

Exemplifiquem com l'usuari entra una comanda E1(x)P, utilitzant la referència "DOBLAR" per tal de que el casc de la línia anterior no es consideri en estoc dins la seva botiga (per què l'usuari ha detectat que l'estoc del sistema i el real no quadren):

Referència	...	...	Unitats	...
C.ARA.DSC	...	...	1	...
DOBLAR	...	...	1	...

## Programació del procés

Hora	Periodicitat
06:30	Dijous de cada setmana

### 5.4.2 Aplicació “Servir Todo Tiendas” (Bloc 1 – Logística)

#### Objecte de procés

De forma resumida, aquesta aplicació tracta totes les comandes que s’han generat:

- A través de la pàgina web comercial de l’empresa.
- A través de la línia d’atenció telefònica.
- A través de la botiga, però que el client desitja que el material li sigui enviat directament al seu domicili (quan hi hagi disponibilitat).

En concret, les comandes que ha de tractar aquesta aplicació, són totes aquelles les quals:

**( Sèrie = E(1X)P ) ^ ( Magatzem = E13 )**

On **E1XP** és el número identificador de la sèrie de documents associats a la central: E13P, E15P.

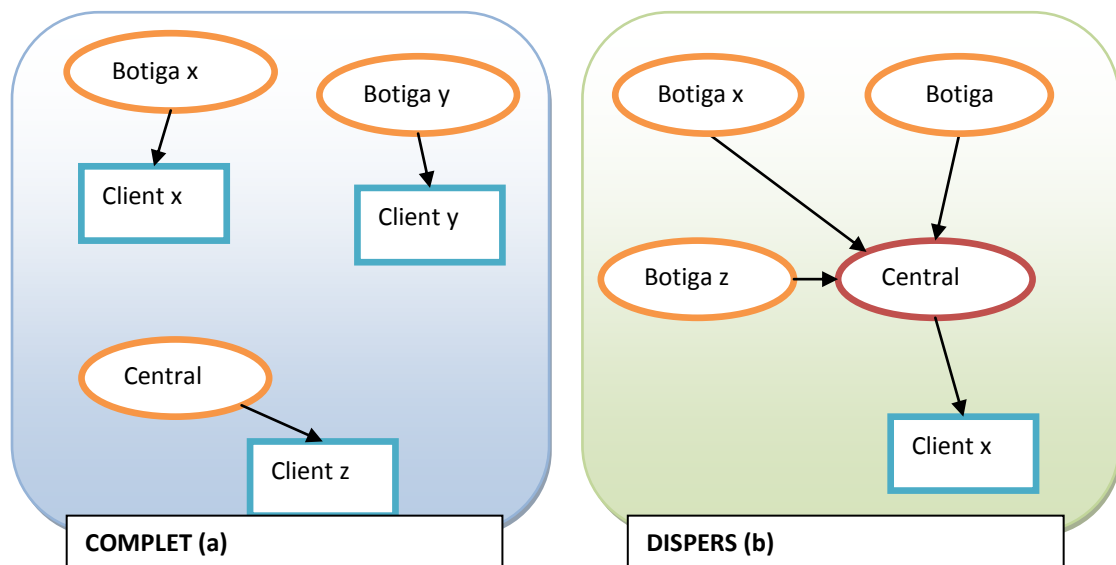
On **E13** és el magatzem central.

#### Descripció del procés

Durant tots els dies de la setmana, des del magatzem central, es processa i controla totes les comandes que s’han d’enviar directament al client final. Hi ha dos possibles escenaris quan parlem de servir material directament al client:

- Completo:** Totes les comandes que tenen tot el material en alguna de les botigues o dins el magatzem central, s’envia directament cap al client des del punt de disponibilitat completa.
- Dispers:** Si una comanda disposa d’estoc per servir-la però l’estoc està distribuït en diferents magatzems de botigues i/o el central, aleshores es fa enviar el material des de les botigues que en disposen cap al magatzem central, on una vegada rebut el material s’enviarà directament al client.

Anotem a continuació uns diagrames conceptuals d'aquests dos escenaris:



### Periodicitat

Aquests procediments en la gestió de l'enviament de comandes al client final, s'utilitzen amb la següent planificació temporal:

- a) Diàriament en el cas del COMPLET
- b) Dilluns, dimecres i divendres en el cas del DISPERS.

### Criteris i parametrització

Actualment els criteris en la programació d'aquesta aplicació estan integrats en el procés de codificació, així com tota possible parametrització. És per aquest motiu que farà falta redissenyar al complet aquests criteris i parametritzacions. Tot i que en l'apartat d'ampliacions de funcionalitats ho anotarem detalladament, us donem un breu resum d'alguns dels criteris que caldrà redissenyar:

- Preferències d'enviament de material per ubicació geogràfica.
- Balanceig d'estocs
- Preferències de magatzems
- Disponibilitat dels magatzems
- ...

**Funcionament intern del procediment DISPERS:**

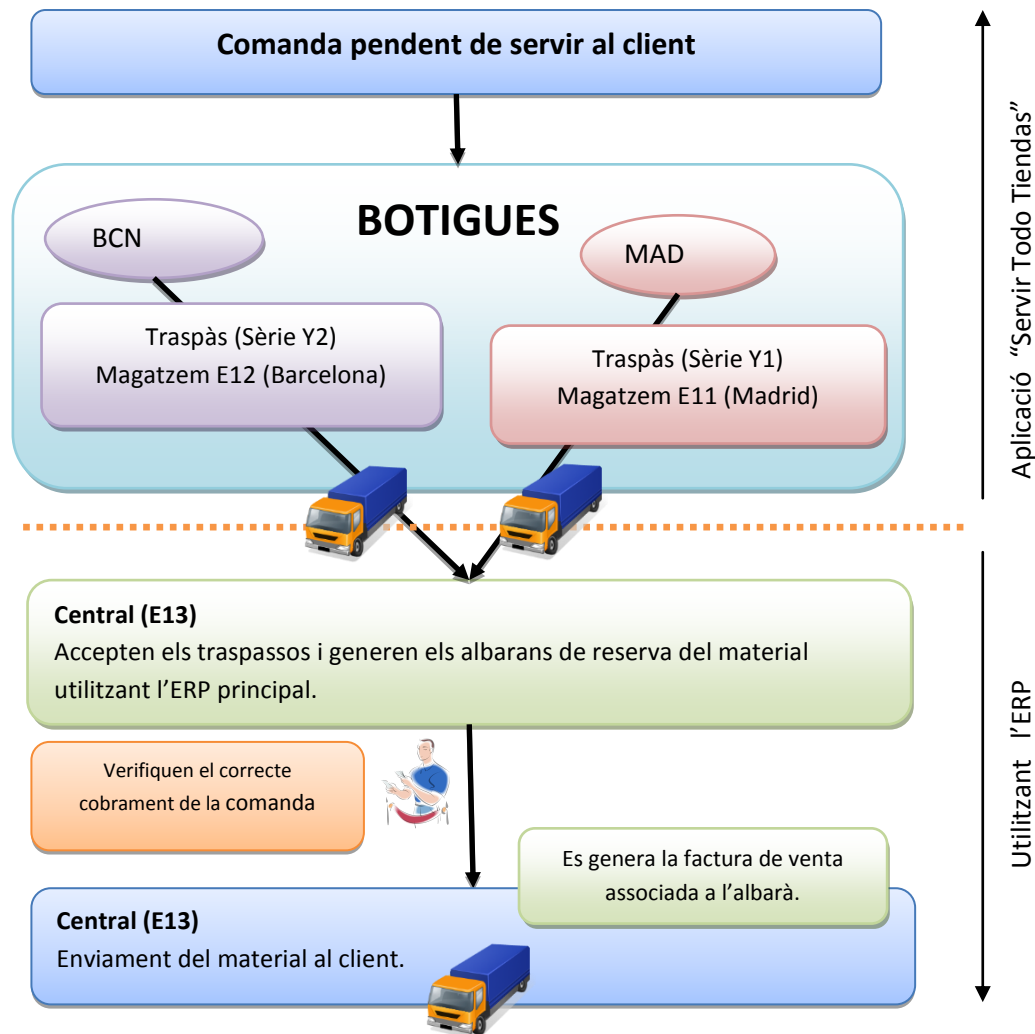
És important identificar que aquest procediment s'executa una part a través de l'aplicació "Servir Todo Tiendas" en qüestió mentre que la resta s'executa utilitzant les eines disponibles de l'ERP:

A través de l'aplicació "Servir Todo Tiendas"

1. Quan el procés revisa les comandes que només es poden servir dispersament, genera albarans de traspàs del material de les diferents botigues per enviar cap a la central.
2. Les diferents botigues reben la notificació d'aquests traspassos de material que hauran d'enviar cap a la central.
3. Les botigues serveixen el traspàs o en cas de falta d'estoc avisen a la central de la falta de disponibilitat (degut a un error de l'estoc en l'ERP, estoc en demostració, defectuós, ...).
4. Es genera un llistat on apareix tot el material que rebran de cada botiga i a quina comanda de client va assignat.

A través de l'ERP:

5. A la central reben el material de les diferent botigues i accepten els traspassos perquè el material quedi computat dins el magatzem central i descomptat definitivament dels magatzems de les botigues d'on prové.
6. Es generen els albarans per reservar el material pel client utilitzant el sistema natiu integrat dins l'ERP.
7. Es reserva el material associat a les diferents comandes.
8. Es verifica l'estat correcte del pagament de les comandes reservades.
9. Es generen les factures de venda, les etiquetes d'enviament i s'envia el material al client.



### Funcionament intern del procediment COMPLET:

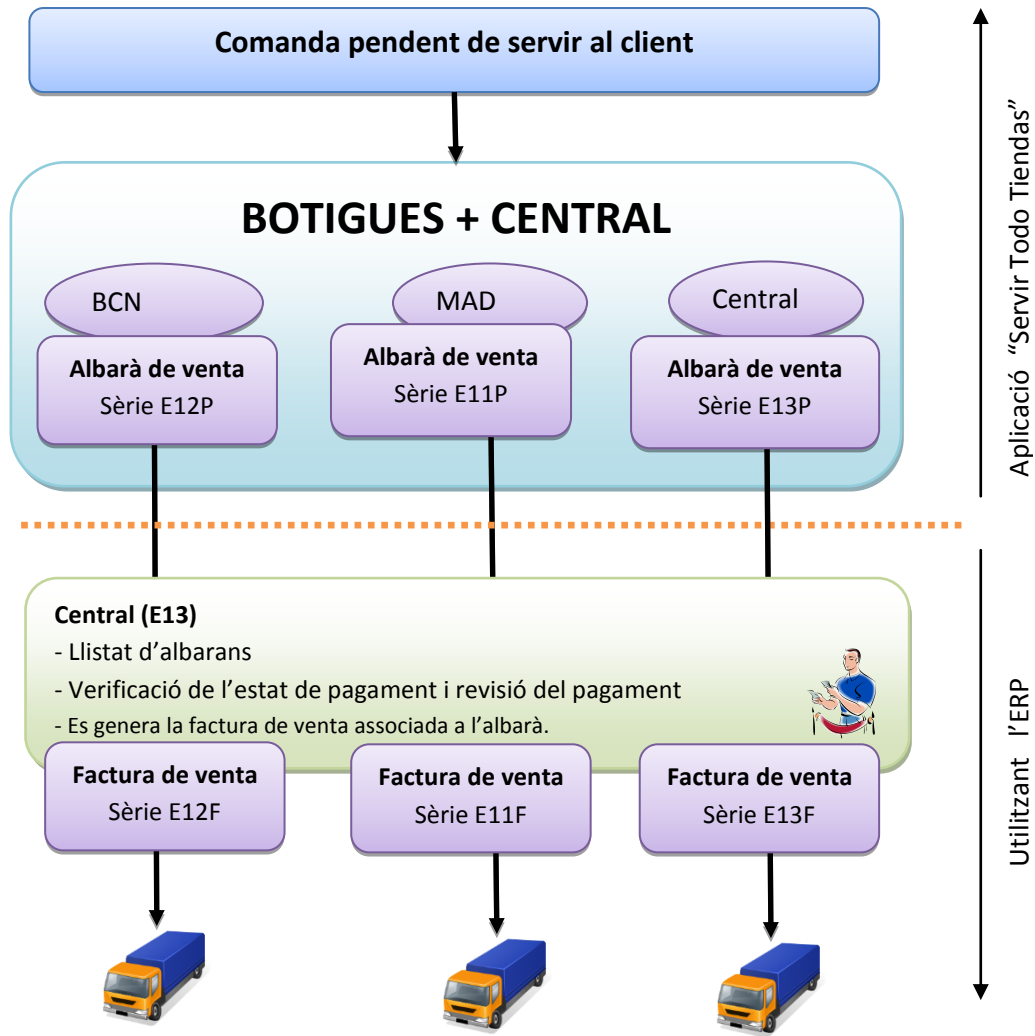
Com en el cas anterior, es genera una part a través de l'aplicació especialitzada, i una altra des del propi ERP.

A través de l'aplicació "Servir Todo Tiendas":

1. Es genera l'albarà de la comanda que té disponibilitat completa, ja sigui en el magatzem central o bé des d'una botiga.
2. Es genera un llistat dels albarans acabats de generar, desglossat per magatzem.

A través de l'ERP:

3. S'imprimeixen tots els albarans, un a un, en base del llistat anterior.
4. Se'n verifica el correcte estat del pagament.
5. Es generen les factures de venda, les etiquetes d'enviament i s'envia el material al client.



### Temporització cíclica dels processos

Per entendre tot el cicle complet d'aquest procés cal tractar-lo des de l'òptica de 2 dies consecutius (Dia 1, Dia 2), on veurem que i quan s'executa. Cada execució de l'aplicació "Servir Todo Tiendas" generarà un seguit d'albarans i traspassos que denotarem de la forma {AlbaransX}, {TraspassosX}.

L'hora indicada és aproximada i indicativa, ja que depèn una mica del volum de material a preparar, possibles incidències que sorgeixen, disponibilitat dels treballadors, ...

Hora	Dia 1	Dia 2
07:30	Verificació de l'estat del pagament. Si tot està correcte, generen una comanda dins de l'ERP.	
8:00	Execució Complet: {71Albarans1} Execució Dispers: {Traspassos1}	Execució Complet: {Albarans3}
08:30	Llistat de caixes per preparar des del magatzem central pel material de {Albarans1}	Llistat de caixes per preparar des del magatzem central pel material {Albarans3}
09:00	Les botigues i central preparen els albarans que els hi pertoqui de la llista {Albarans1}	Les botigues i central preparen els albarans que els hi pertoqui de la llista {Albarans3}
		La central prepara el material de la llista de caixes del dia anterior dels {Albarans2}
		S'envia el material dels {Albarans2} cap al client.
10:00	Les botigues preparen el material de {Traspassos1} per enviar cap a la central.	
13:00	Les botigues envien el material dels {Albarans1} cap al client i el material de {Traspassos1} cap a la central.	
15:00	La central accepta els traspassos {Traspassos1} de material provinent de les botigues.	
15:30	Generen els albarans de venda {Albarans2} pel material de {Traspassos1} des de l'ERP, tot i que el material no arribarà físicament fins el Dia 2.	
16:00	Llistat de caixes per reservar el material dels {Albarans2}	

Aquest cicle de vida de dos dies es reparteix de forma continua durant la setmana, seguint el patró següent:

Dilluns	Dimarts	Dimecres	Dijous	Divendres	Dissabte
Dia 1	Dia 2	Dia 1	Dia 2	Dia 1	Dia 2

### 5.4.3 Aplicació “Pendientes” i “Pendientes Y3” (Bloc 1 – Logística)

Existeixen dos aplicacions per a fer un control i gestió de les comandes pendents de servir al client, depenent de si aquestes comandes s’han d’enviar directament al client final o bé si són comandes que recolliran a alguna de les botigues existents.

Primer de tot, definim una mica més acuradament que ha d’entendre el sistema per a **comanda pendent**:

Una comanda pendent és tota aquella que conté alguna línia referent a un article, la qual té unitats pendents d’entregar al client. Per això cal anotar que en les taules on s’enregistren les línies de cada comanda existeixen tres tipus d’unitats:

- **UNIDADESTOTAL** – N° total d’unitats que s’han de servir al client.
- **UNIDADESREC** – Unitats ja entregades al client (i per tant s’ha fet un albarà, tiquet o factura)
- **UNIDADESPEN** – Unitats pendents d’entregar al client (no s’ha fet albarà, ni tiquet ni factura)

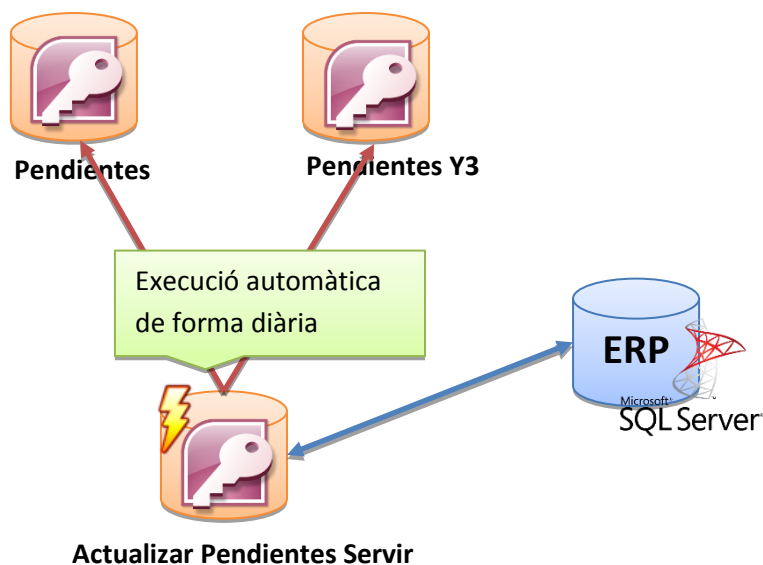
*Per tant tota comanda, que tingui alguna de les línies associades amb les UNIDADESPEN>0, es considerarà una comanda pendent.*

Vist aquest aclariment tenim que, de les dos aplicacions:

- **“Pendientes”**: S’encarrega de visualitzar les comandes pendents d’enviar directament al client final.
- **“Pendientes Y3”**: S’encarrega de visualitzar les comandes pendents de recollir pel client a les diferents botigues, ja sigui perquè manca estoc o bé perquè el client encara no les ha recollit.

Aquests processos s’actualitzen de forma diària (a través d’un procés automàtic) que actualitza l’estat de les comandes pendents afegint les noves que es detectin, eliminant les que ja estan servides en la totalitat o actualitzant les que encara estan pendents.





### Funcionalitat d'ambdós aplicacions

Tot i que l'objectiu d'ambdós aplicacions explora comandes de naturalesa diferent, comparteixen les mateixes funcionalitats:

- Llistar les comandes i el material pendent en concret, així com les dades de contacte del client.
- Gestionar notes i comentaris a nivell intern pel seguiment, i informació de control sobre la comanda.
- Modificar la data prevista d'arribada del material i notificar canvis en aquesta data als usuaris.
- Notificar la necessitat de trucar al client (nacional o internacional).
- Llistar les comandes de proveïdor per l'article de la comanda pendent en qüestió, per obtenir informació extra sobre la previsió o estat de l'arribada de material.
- Unitats pendents de l'article de la comanda pendent, en altres comandes pendents d'altres clients.
- Cercar comandes pendents per un client en concret.
- Generar consultes directament al magatzem des de les botigues (sense existir una comanda associada) per conèixer la disponibilitat d'un article, present o futura.

També se'ns permet la generació de llistats per:

- Llistats de control per poder generar informes de les trucades pendents d'efectuar a clients i alarmes d'arribada de material (per verificar-ne l'exactitud i poder actuar amb antelació en la comunicació amb el client).
- Llistat de clients en espera, que tenen comandes pendents de servir, ja sigui per falta d'estoc o per que estem esperant la recollida del material a la botiga.
- Llistat de material que té associades faltes d'estoc.
- Llistat dels clients ja servits (es pot anar buidant aquesta taula històrica).

## 5.5 Contrastar les funcionalitats vigents amb el client

Després de l'etapa de recollida de funcionalitats vigents en el sistema, establim les corresponents reunions amb el client, entregant-li el corresponent informe fruit del punt 5.4, per tal de:

- Revisar meticulosament les diferents aplicacions
- Revisar-ne altre cop la completesa i vigència.
- Determinar noves necessitats en el sistema.
- Detectar necessitats actualment acomplertes però necessàries de revisió.

D'aquest punt en sorgeix el punt 5.6 que consisteix en el detall de les noves funcionalitats i rectificacions a realitzar en les aplicacions existents.

## 5.6 Ampliació de funcionalitats per part del client

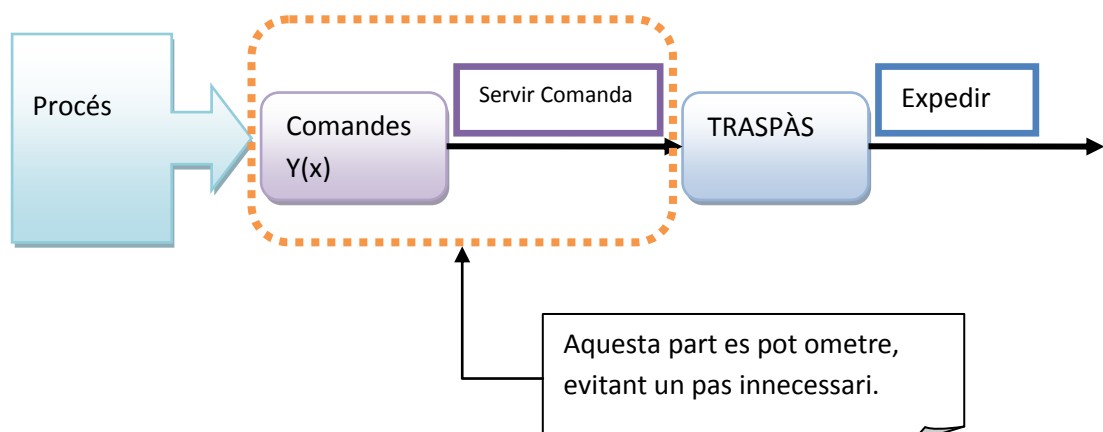
Com en l'apartat 5.4, ens centrem exclusivament en les aplicacions més important i interessants per l'anàlisi.

### 5.6.1 Aplicació "Pedidos" (Bloc 1 – Logística)

#### Canvi en la periodicitat i el flux de treball (*Workflow*) de les botigues i el magatzem central

El client vol modificar el paradigma de treball de l'aplicació per complir els següents requisits:

- No s'executi el procés un sol cop a la setmana, sinó dos cops al dia. Això permetrà que sempre es reservi el material per les botigues amb comandes pendents, el més aviat possible i evitar que comandes o compres posteriors consumeixin estoc que pertocaria a una comanda preferent per antiguitat.
- L'enviament de material es seguirà fent un cop a la setmana (minimitzar costos de transport), però la diferència és que durant la setmana s'hauran anat acumulant articles per reservar entre les botigues en diferents traspassos acumulatius. (Un traspàs acumulat per cada combinació de botiga origen – botiga destí). Això farà que l'estoc dels articles quedi reservat per les botigues que el demanden i per tant no es puguin servir a comandes posteriors.
- El llistat automàtic que actualment s'envia a les botigues, s'hauria d'enviar a l'encarregat en format de full de càlcul (*Microsoft Excel*). Per la resta de treballadors de la botiga hauria d'existir una interfície còmoda i senzilla per poder visualitzar en tot moment el material que han de reservar per les botigues i el que rebran durant aquesta setmana.
- Actualment aquesta aplicació genera comandes entre les botigues, cosa que implica que posteriorment han de transformar aquestes comandes generades a les botigues, en albarans de venta interns (traspassos). Es vol saltar aquest pas intermedi i generar directament els traspassos, evitant un pas extra a nivell de gestió per part de les botigues i que a la vegada sovint indueix a errors que podríem evitar.



- Cal agilitzar el procés de gestió de l'enviament del material associat a les comandes que les botigues fan al magatzem central, de forma que quan el material arriba al proveïdor, aquest mateix sistema sigui capaç de detectar aquestes comandes pendents i servir-les primer de tot (ja que són les comandes amb més antiguitat). Aquesta part també ha de formar part de l'execució continua de l'aplicació per assegurar que es minimitza el temps d'espera del client.

### **Objecte del procés**

- Actualment el sistema només tracta comandes pendents dins l'interval d'una setmana. Això provoca que comandes anteriors als 7 dies que utilitza en la cerca no es processin adequadament. L'aplicació ha d'analitzar en cada execució totes les comandes amb unitats pendents, ja que pot haver aparegut estoc en altres botigues (regularitzacions, devolucions, compres a proveïdors,...) o bé en el magatzem central (generalment compres a proveïdors).

### **Canvis en la gestió de faltes d'estoc**

Es vol eliminar l'ús de la tècnica del "DOBLAR" i que es puguin registrar de manera controlada les faltes d'estoc associades al magatzem d'una botiga o al central, perquè la resta d'aplicacions del sistema puguin tenir-les en compte en el càlcul de l'estoc real disponible dels articles. Aquest sistema de gestió de faltes haurà de permetre:

- Enregistrar faltes d'una manera àgil i còmoda des de les pantalles de control del material a reservar per les altres botigues o el magatzem central.
- Les dades que caldrà enregistrar en un falta són:
  1. Data i hora creació
  2. Usuari que la registra
  3. Motiu de la falta
  4. Data i hora de la solució
  5. Usuari que la soluciona
  6. Motiu de la solució.
- Poder llistar les faltes existents, i filtrar-les pel seu estat i magatzem d'origen.
- L'administrador o encarregat de les botigues, haurà de ser l'únic capaç d'eliminar faltes de forma manual. Els usuaris no administradors simplement podran entrar noves faltes en el sistema des de les pantalles corresponents o de forma manual des de la pantalla destinada per aquest objectiu.

### Prioritats i disponibilitats dels magatzems

- Cal dotar a l'aplicació de la capacitat de gestionar i modificar fàcilment les prioritats entre botigues pel que fa a la decisió dels moviments de material. Adjuntem a continuació la taula de prioritats desitjada i que caldria poder editar de forma àgil. Actualment no compleix aquesta estructura, però caldria que ho fes:

Magatzem d'on surt el material												
	E11	E12	E14	E15*	E16	E17	E18	E19	E1Z	E13	A11	A12
E11		7	6		4	3	5	2	8	1	9	10
E12	7		5		4	3	2	6	8	1	9	10
E14	8	7			3	2	4	5	6	1	9	10
E15	7	11	5		4	3	2	6	8	1	9	10
E16	8	7	3			2	4	5	6	1	9	10
E17	8	7	2		3		4	5	6	1	9	10
E18	8	7	4		3	2		5	6	1	9	10
E19	8	7	5		3	2	4		6	1	9	10
E1Z	8	7	6		3	2	4	5		1	9	10

(\*E15 – Magatzem virtual associat a la WEB)

- Cal permetre activar o desactivar temporalment el còmput d'algun dels magatzems de les botigues, en cas de festivitat local, per evitar que es demani material que arribaria amb demora. Pot ser que un article estigui present en diferents magatzems i es demani en un que no roman obert durant aquest dia. Segons el dia de la setmana en que es produeixi aquesta casuística, pot ampliar una setmana el temps d'entrega al client.

### Consideracions en el càlcul d'estoc disponible

L'aplicació ha de tenir en compte en tot moment, les unitats disponibles realment a la botiga. El sistema ERP principal ens informa de cada article i per cada magatzem de:

- Unitats pendents d'entregar (SE): Ens indica les unitats que hi ha pendents d'entregar al client final (ja sigui perquè no hi ha estoc o perquè estan pendents de recollir o entregar)
- Unitats en trànsit (ST): Ens indica la quantitat d'articles que estan en traspàs d'altres magatzems cap al nostre. Aquests traspàsos són referents a material enviat des de les botigues o la central cap al nostre magatzem.

- Unitats demanades a proveïdor (SP): Ens indica la quantitat d'articles demanats al proveïdor i pendents de rebre en estoc.
- Estoc físic (SF): Fa referència a l'estoc físic de l'article, sense tenir en compte si existeixen reserves, material que està en camí o material demanat al proveïdor.

Per tant les unitats realment disponibles, o estoc real (SR), que es poden enviar a altres botigues, han de respectar el càlcul de les diferents variables anotades i es pot expressar en la següent formula:

$$SR = SF + ST - SE$$

$$\text{<Estoc Real>} = \text{<Estoc Físic>} + \text{<Estoc en Transit>} - \text{<Estoc pendent d'entrega>}$$

**Nota:** El client no vol tenir en compte o considerar en aquest moment les unitats pendents de rebre del proveïdor, ja que això pot suposar retards en l'entrega de les comandes al client degut a la falta d'una data factible d'entrega del proveïdor a la pròpia empresa.

### Implicacions en altres aplicacions

Caldrà dotar a l'aplicació encarregada de la gestió, visualització i control de les comandes pendents, de forma que pugui disposar d'una consulta ràpida del seguiment de la comanda d'un client, per millorar la comunicació amb el client en casos d'anomalies i millorar l'anticipació a conflictes, incrementant el feedback entre el client i la botiga o la central.

### Gestió retroactiva

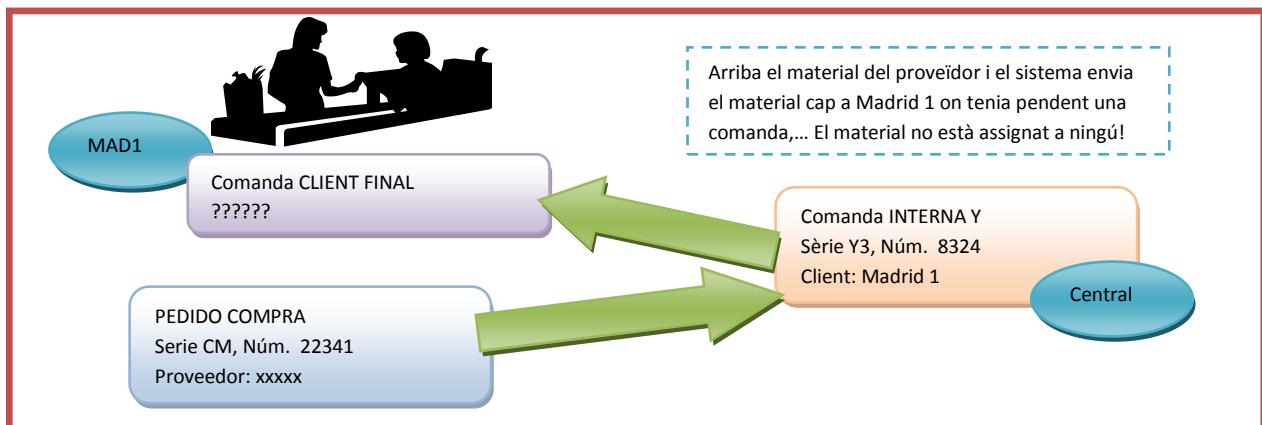
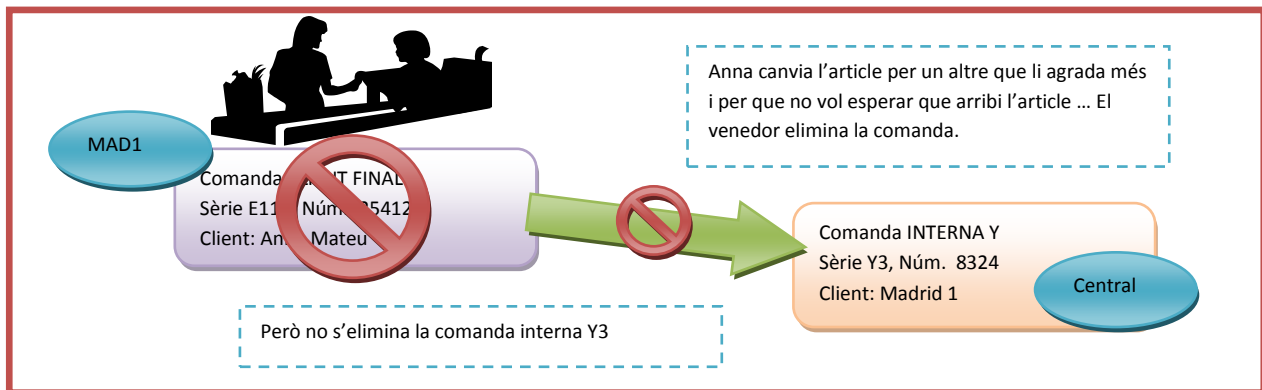
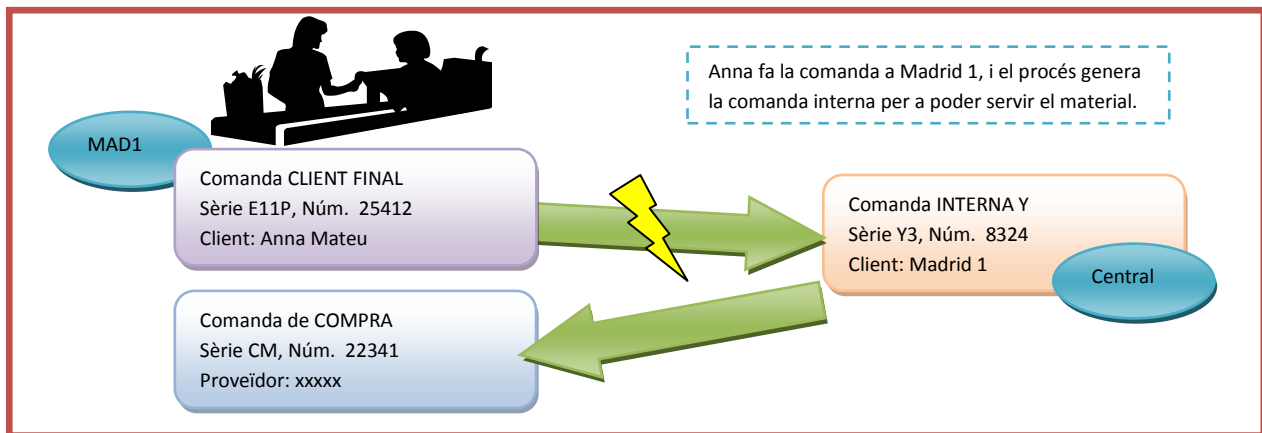
Sovint les botigues eliminen la comanda feta pel client final per diferents motius (el client no vol esperar més, vol canviar l'article, ...). En aquests casos queden descontrolades les comandes que en el seu moment s'havien generat a la central perquè s'encarregués de gestionar-ne la disponibilitat.

Quan arriba el material que s'ha demanat al proveïdor per aquesta comanda, s'acaba enviant a la botiga que l'havia demanat, sense necessitat real:

- Generant un sobre estoc innecessari.
- Generant uns costos de transport innecessaris.

El sistema hauria de ser capaç de controlar aquest escenari, i abans d'enviar el material associat a una comanda d'una botiga, verificar realment de la necessitat real d'estoc d'aquesta contrastant les unitats que té pendents, les que té en estoc real i les faltes que puguin existir.

Intentem exemplificar aquest escenari conflictiu mitjançant els gràfics següents:



### Altres funcionalitats desitjades

- Seria necessari afegir un sistema d'avísos en línia, per recordar a les botigues els traspassos que tenen pendents d'acceptar i per tant d'incorporar en el seu estoc, ja que es detecta que en diverses ocasions es queden pendents en el sistema i provoquen problemes d'estocs al sistema.

### 5.6.2 “Servir Todo Tiendas” (Bloc 1 – Logística)

El client desitja una remodelació total d’aquesta aplicació per:

- Ajustar-lo a les necessitats reals de la lògica de negoci de l’empresa
- Evitar la dependència d’usuaris experts en el procés.
- Minimitzar el temps d’execució global.
- Fer-la més flexible i capaç de configurar-se amb uns paràmetres mínims.

#### Canvi en la periodicitat i el flux de treball (*Workflow*) de les botigues i el magatzem central

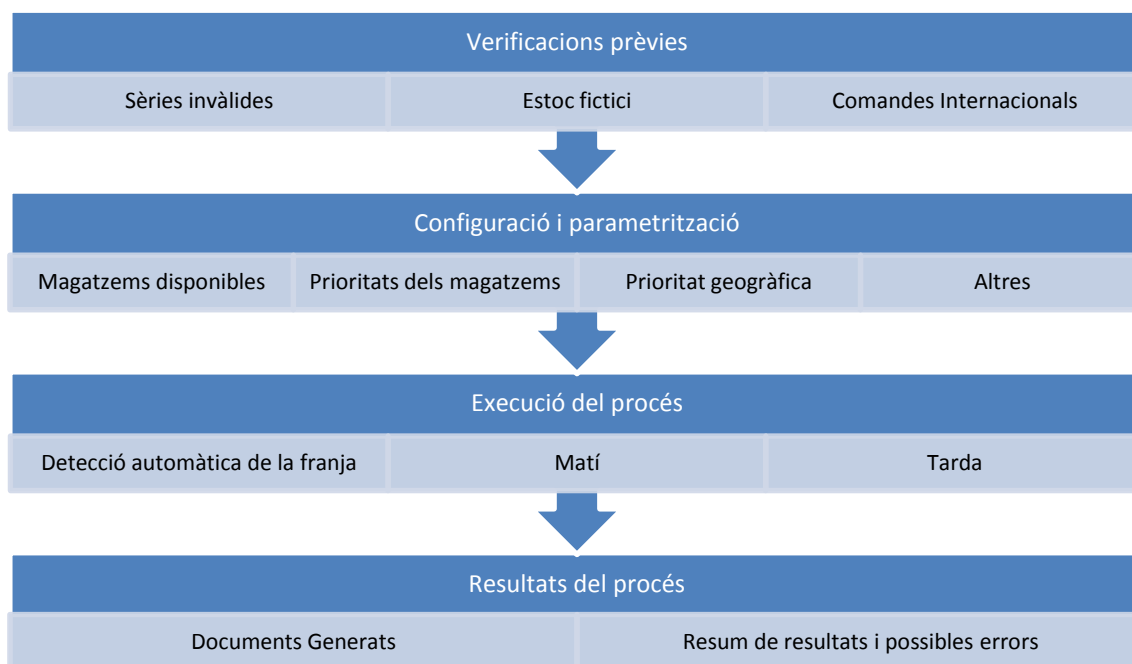
El client vol modificar el paradigma de treball de l’aplicació, de manera que compleixi els següents requisits:

- **Temporització:** No s’executi el procés en blocs de 2 dies (Dia1 + Dia2) sinó que s’executi de forma diària. En concret es vol plantejar dos esquemes especialitzats de funcionament, un pensat per l’execució durant el matí i un per la tarda, tot i que el sistema hauria de detectar-ho de forma automàtica en base de l’hora de l’execució.
- **Unificar el procés:** Actualment, i com hem vist en l’anàlisi de l’aplicació, el flux de treball està dividit en l’ús de dos aplicacions (ERP + Aplicació “Servir Todo Tiendas”). Es desitja que una sola aplicació s’encarregui del màxim possible de tasques dins del procés i unificar així la feina i reduint el temps d’execució i dependència manual. Hi ha un seguit de tasques que es poden automatitzar, però d’altres que hauran de seguir elaborant-se de forma manual.
- **Configuració:** Dotar a l’aplicació de la configuració necessària per poder parametritzar el procés en base d’uns criteris que analitzarem tot seguit.
- **Balanceig de l’estoc:** Es vol fer un ús intel·ligent del consum d’estoc disponible als diferents magatzems.
- **Automatitzar:** Actualment l’aplicació “Servir Todo Tiendas”, va acompanyat d’altres aplicacions que ajuden a finalitzar el procés (com per exemple el llistat de caixes) i altres procediments dins l’ERP (impressió dels albarans). Es vol automatitzar el màxim aquest procés de manera que s’executi tot en bloc i generi un resultat final amb poca interacció de l’usuari.
- **Generació de documents de suport:** Quan s’executi l’aplicació de “Servir Todo Tiendas” de forma automatitzada, es desitja que es generin els corresponents llistats de control i actuació per què les persones assignades a les diferents tasques puguin procedir amb la seva labor. Es voldrà assignar als diferents usuaris del sistema un seguit de documents que tenen permís per visualitzar i imprimir, de forma que cadascun d’ells només vegi els que li pertoca.



- **Control de sèries invàlides:** Degut a errors humans, de vegades s'entren comandes dins del sistema utilitzant sèries documentals inadequades. El sistema actual les tracta igualment generant resultats incoherents a nivell organitzatiu. Per tant es vol dotar al sistema d'un llistat de control de sèries invàlides abans d'iniciar l'execució del procés funcional.
- **Estoc fictici:** L'empresa utilitza un seguit d'articles per la venda a través de la pàgina web, que en realitat ja estan obsolets i sense estoc, però es poden assignar articles d'identiques prestacions sense cost extra pel client. Actualment aquests articles s'han de verificar manualment i per tant suposa una tasca força tediosa. Es vol integrar aquest sistema de control i canvi automàtic dels articles associats des del mateix procés.
- **Revisió de comandes internacionals:** L'enviament de comandes internacionals es fa des d'una empresa de transport diferent a les nacionals, i amb costos diferents. Justament per això és molt important que en el procés d'enregistrar les comandes s'hagi assignat correctament la nacionalitat del client. Actualment han d'alternar la consulta de l'aplicació "Servir Todo Tiendas" amb la de l'ERP per corregir possibles errades en la introducció de dades. Es vol integrar tot aquest procés també en l'aplicació "Servir Todo Tiendas", de forma que des de la mateixa aplicació pugin corregir les dades de contacte del client sense necessitar d'accedir a l'ERP.
- **Control i seguiment des de les botigues:** Es vol dotar d'un sistema unificat per la revisió i preparació del material de les botigues que han d'enviar cap a la central. En concret es vol unificar amb el mateix sistema que ja volen integrar per l'aplicació "Pedidos", de forma que des d'una sola pantalla puguin gestionar tant el material que han de reservar per les altres botigues (enviament setmanal), com pel magatzem central (enviament diari)
- **Extracció de material de traspassos d'estocatge:** Durant el cap de setmana es preparen un seguit de traspassos de material de la central cap a les botigues per emplenar l'estoc d'aquestes. Si es detecta un article necessari per a poder servir la comanda d'un client, el sistema hauria d'extreure aquest material del traspàs i assignar-lo a la comanda del client.
- **Execució controlada i independent:** Per tal d'evitar que el sistema quedi paralitzat per un possible error que es generi al tractar una comanda dins el conjunt de comandes a tractar, es vol que el sistema sigui capaç d'executar les comandes de forma independent. Si alguna d'elles produeix algun error, es recapitularà i es mostrarà en un resum final a l'usuari, però mai s'ha de paralitzar la resta de possibles comandes a servir.

Tenint en compte tots aquests punts anotats podem anotar a continuació un diagrama del flux de treball que haurà de dur a terme l'aplicació:



### Verificacions prèvies

- **Sèries invàlides:** Es llistaran els documents amb sèries invàlides i s'exclouran de l'objectiu de l'aplicació. L'usuari haurà de corregir-les manualment.
- **Estoc fictici:** Es llistaran els articles categoritzats com d'estoc fictici, i es permetrà a l'usuari assignar l'article substituït a través d'una cerca dins dels articles existents. Caldrà poder gestionar quins són els articles que anomenen "d'estoc fictici" i poder assignar l'article substituït per defecte.
- **Comandes internacionals:** Es llistaran totes les comandes considerades internacionals (es detectarà pel camp "PAIS" i "PROVINCIA" de la base de dades de l'ERP), perquè l'usuari pugui verificar si tots els que hi apareixen són realment internacionals. Si convé l'usuari podrà modificar les dades de la direcció del client per ajustar-les a la realitat o corregir-les.

### Configuració i parametrització

- **Magatzems disponibles:** Caldrà poder escollir quins magatzems entren dins el procés de càlcul, tant per la part del tractament de disponibilitat COMPLETA com la DISPERSA.
- **Prioritats dels magatzems:** Caldrà especificar una cadena de prioritats a l'hora d'escollir de quin magatzem extreure el material en cas de múltiple disponibilitat (interessa deixar abans amb menys estoc certs magatzems que d'altres).
- **Prioritat geogràfica:** Caldrà crear un sistema ajustable pel que fa a les prioritats d'enviament d'una comanda COMPLETA. Volem poder decidir per cada país i província, quina cua de prioritats té a l'hora d'escollir la disponibilitat d'estoc en múltiples magatzems.
- **Altres:** Cal poder ajustar altres elements en la configuració com poden ser: Recàlcul d'estocs prèvia execució del procés, considerar unitats en trànsit al consultar les disponibilitats, extreure material de traspassos d'estocatge,...

**Execució del procés:**

L'execució del procés després de les verificacions prèvies, requerirà una sola interacció de l'usuari, i es calcularà de forma automàtica la franja d'execució que li pertoca (matí o tarda), permetent però la selecció manual d'aquesta.

**Resultats del procés:**

- **Resum de resultats del procés:** Després de la finalització del procés, caldrà mostrar a l'usuari el resum de resultats (quantes comandes s'han servit, traspassos generats, ...) i qualsevol possible error que s'hagi generat durant l'execució (indicant en quina comanda, codi i missatge d'error).
- **Documents generats:** El procés genera els documents necessaris per repartir les següents tasques distribuïdes de l'empresa.

Els documents que cal generar, són:

- Resum d'albarans generats.
- Llistat d'albarans generats (un full per cada albarà en idèntic format que els utilitzats en l'ERP).
- Llistat de caixes a preparar al magatzem.
- Llistat de material extret de traspassos d'estocatge.
- Llistat detallat d'albarans amb disponibilitat parcial (només hi ha una part del material disponible).
- Llistat de canvis d'articles d'estoc fictici que s'han corregit, amb la informació de la comanda, telèfon del client, ...
- Llistat de traspassos generats de les botigues cap a la central, amb els corresponents articles involucrats.
- Llistat d'albarans amb comentaris o personalitzacions (s'utilitzen uns codis d'article especials per aquesta funció) per a revisar abans de l'enviament, ja que pot suposar contactar amb el client per qualsevol aclariment.
- Llistat d'articles a preparar al magatzem per a totes les comandes, ordenats per la ubicació física de l'article, per permetre la recollida del material d'una forma òptima i àgil.

**Faltes d'estoc**

Igual que en la primera aplicació "Pedidos" vista anteriorment, actualment s'utilitza l'ús d'una referència anomenada "DOBAR" per a indicar la falta d'estoc d'algun article. Com en el cas anterior, haurem d'utilitzar un sistema de faltes que en permeti l'adequada gestió i control d'aquestes.

**Balanceig d'estocs**

Al escollir d'on hem d'extreure el material per a servir les comandes al client, haurem d'aplicar un algorisme que respecti el balanceig d'estocs, de forma que apliqui un consum d'estocs distribuït, evitant deixar un magatzem sense quan n'existeix disponibilitat sobrant en d'altres.

### 5.6.3 Aplicació “Pendientes” i “Pendientes Y3” (Bloc 1 – Logística)

#### Canvis en l'estructura de les aplicacions

El client planteja dos objectius primordials en l'ús d'aquestes dos aplicacions:

- **Unificar-les:** Crear una sola aplicació que permeti consultar indistintament les comandes que s'han d'enviar a client final, així com les que es recullen directament a la botiga. Per tant de la suma d'aquestes dos aplicacions n'apareixerà una de sola anomenada **“Pendientes de Servir”**
- **Dades actualitzades en tot moment (on-line):** Fins ara ambdós aplicacions contenien dades actualitzades de forma diària (Aplicació “Actualizar Pendientes Servir”), o sigui eren *“off-line”*. Es desitja que totes les dades siguin completament síncrones amb les bases de dades de l'ERP i per tant estiguem visualitzant les dades actuals al moment (*on-line*).

#### Ampliació de funcionalitats

A part de les funcionalitats existents en l'aplicació actualment, es vol dotar a la nova aplicació de funcionalitats addicionals:

#### Indicadors d'estat de la comanda

Caldrà fer ús d'uns indicadors visuals de ràpida interpretació que ens permeti tenir informació global de la comanda i de les seves línies de forma independent també. En concret volem disposar d'indicadors per diferents categories:

- Estat de la comanda: Disponibilitat parcial, total o sense disponibilitat.
- Estocs en trànsit: Existeixen unitats de camí (en transport)
- Comandes fetes al proveïdor: Existeixen unitats demanades al proveïdor o al magatzem central.
- Consultes realitzades sobre la comanda: Consultes associades a la interacció amb el client o internament per comunicar dates d'entrega, trucades pendents, notes pendents de llegir,...

#### Llistat de comandes navegable, complet i accessible

Es vol disposar d'un llistat que sigui fàcil, còmode i complet pel que fa a les diferents comandes i de les línies d'articles de les mateixes, de forma que al seleccionar una de les comandes s'accedeixi fàcilment a les seves línies, i una vegada accedim a la línia s'accedeixi també fàcilment a les dades de l'article que la representa.

En concret es demanen les següents funcionalitats en dit llistat:

- Es pugui ordenar pels camps data, número de sèrie, número de document i nom del client.
- Poder imprimir un llistat (detallat o no) del llistat de comandes que s'estigui visualitzant.
- Poder filtrar el llistat actual de comandes per diferents criteris:

1. Per l'estat de la comanda.
  2. Per l'existència o no de consultes associades.
  3. Per un article en concret.
  4. Per la marca d'un article en concret
  5. Segons si s'ha processat o no la comanda a través d'algunes de les aplicacions bàsiques de gestió de comandes ("Pedidos" o "Servir Todo Tiendas").
  6. Per un interval de dates.
- Poder cercar totes les comandes històriques d'un client.

### **Informació agregada a la comanda**

També es vol disposar de fàcil accés a informació extra de la comanda o les seves línies, com pot ser:

- Dades del client
- Existència d'avançaments en el pagament de la comanda.
- Dades de l'article (Sobretot referent a l'estoc)
- Compres pendents de rebre del proveïdor.
- Resum d'estocs de l'article pendent per magatzems.
- Relació d'altres comandes que tenen pendent el mateix article de la que estem visualitzant.

### **Consultes associades a les línies de comandes pendents**

Caldrà poder associar consultes a cada una de les línies de material pendent d'una comanda, o sigui a cada article/s pendent de servir al client.

Quan parlem d'una consulta la considerem com un conjunt de converses on diferents usuaris poden anar anotant observacions i marcant diferents fites informatives en el procés de comunicació amb el client final o en l'estat de la comanda, com poden ser:

- Petició d'informació genèrica.
- Necessitat de trucar a un client.
- Marcar que s'ha trucat al client i s'hi ha pogut parlar.
- Intent de contactar amb el client infructuós.
- Deixat missatge al contestador del telèfon del client.
- Esperant la resposta d'un client.
- Cal anular la comanda.

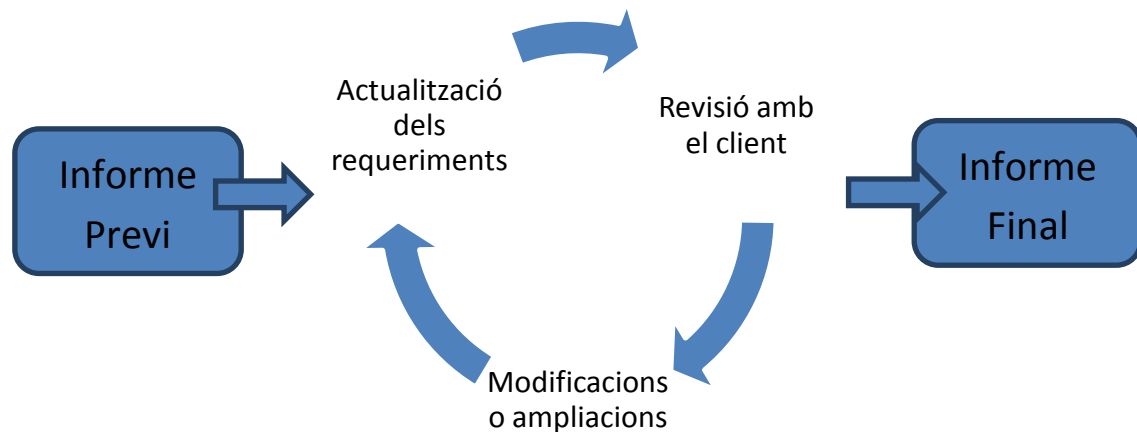
Al crear una entrada o conversa dins la consulta, caldrà enregistrar:

- Data i hora de la nova entrada
- Origen i destí de la conversació (Botiga1 -> Botiga2, Botiga5 -> Magatzem, Magatzem -> Botiga1, ...)
- Usuari que genera la conversa
- Tipus (alguna de les possibles fites informatives anotades anteriorment)
- Si s'ha llegit o no per part del destinatari.

## 5.7 Informe previ i revisió de l'anàlisi de requeriments

Una vegada recaptada tota la informació referent a l'anàlisi de requeriments de les aplicacions existents així com noves funcionalitats que es vulguin implementar, es presenta un informe previ al client amb els continguts de l'apartat 5.6.

S'estableixen les corresponents reunions per a l'estudi de l'informe amb el client, deixant obert el cicle de vida de l'anàlisi fins que es determini en la seva totalitat les necessitats actuals i reals del sistema desitjat.



Una vegada obtingut l'informe final, entrem dins la fase del disseny tècnic i funcional.

## 6. Especificació i disseny

En aquest apartat ens centrarem a especificar i dissenyar l'arquitectura del sistema SetelSys, per complir amb els requisits detectats i desitjats.

Tot seguit analitzem els models de dades existents ja en el sistema actual prèvia introducció del sistema SetelSys i amb els quals el nou sistema s'haurà de comunicar tant per la extracció com introducció de dades.

### 6.1 Models de dades existents en el sistema

Per tal de poder tenir una imatge semànticament fiable de les bases de dades existents en el sistema, tant pel que fa a l'ERP com a la pàgina WEB comercial, hem utilitzat un procés d'**enginyeria inversa** per poder plasmar el model entitat relació existent en les bases de dades actuals de dits components.

Analitzem en tot moment únicament les entitats relacionades amb l'objectiu de la nostra aplicació deixant a part tota la resta d'entitats de la base de dades que no ens incumbeix.

#### 6.1.1 Model de dades de l'ERP principal

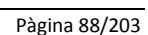
##### Comandes de venda

Les taules centrals d'aquest model són, **PEDVENTACAB** i **PEDVENTALIN**, que s'utilitzen per emmagatzemar la informació de la capçalera i línies de la comanda. Existeixen un seguit de taules fruit de la relació entre alguns dels seus camps i altres taules de l'ERP:

- **SERIESDOC**: Associa el document a la sèrie i tipus concrets per cada element. A la vegada aquesta taula es relaciona amb **SERIES** i **TIPOSDOC**.
- **CLIENTS**: Associa el document a un client concret.
- **ARTICULOSLIN**: Associa la línia del document a un article concret del sistema (Utilitzant la clau primària corresponent – Codi + Talla + Color). A la vegada aquesta taula depèn d'**ARTICULOS** on s'emmagatzema la informació genèrica dels articles (Referència, marca, departament, etc.)
- **ALMACEN**: Associa la línia del document a un magatzem existent. Això és necessari per que el sistema sigui conscient d'on s'extreu el material.

També hi ha un parell de taules que depenen de la PEDVENTACAB:

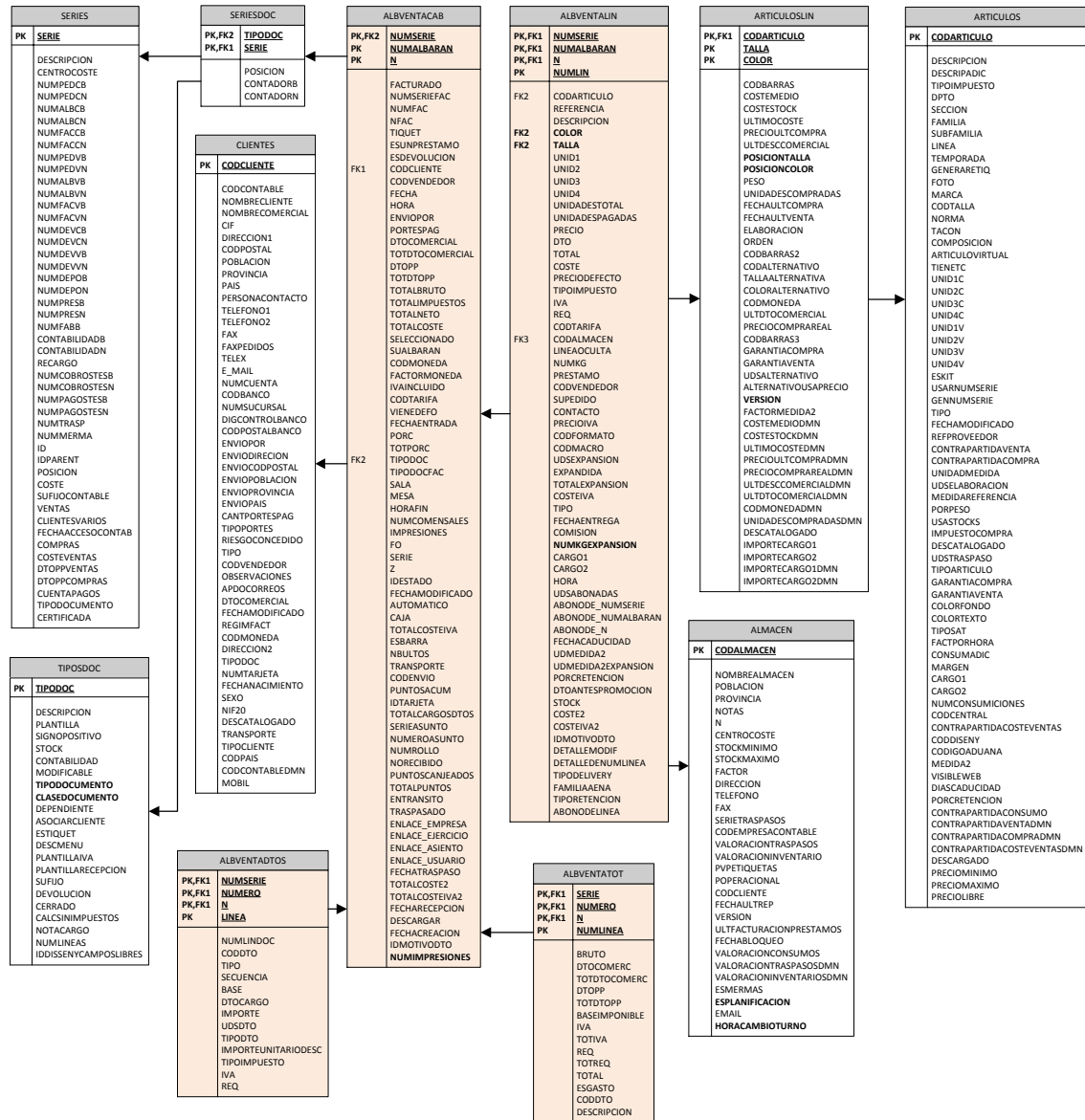
- **PEDVENTATOT**: On s'emmagatzema la informació corresponent als totals de la comanda.
- **PEDVENTADTOS**: On s'emmagatzema informació dels descomptes especials associats al document (descompte comercial, descompte per "pronto pago",...)





# Albarans de venta

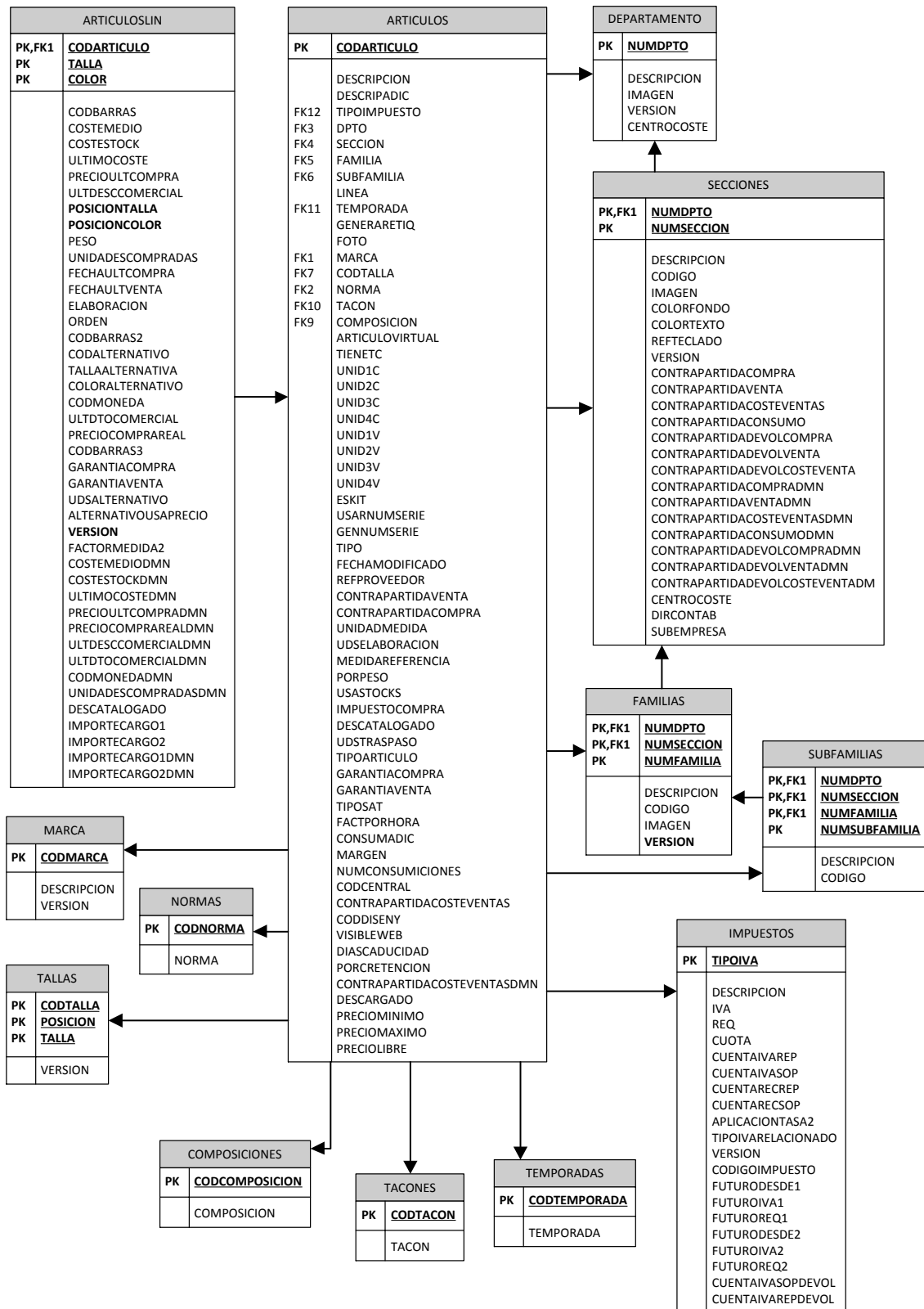
Les relacions en aquest bloc de taules són les mateixes que en les comandes de venta, però utilitzant les entitats relacionades amb els albarans en comptes de les comandes, us les indiquem amb una tonalitat **especial** per visualitzar les úniques diferències.



## Articles

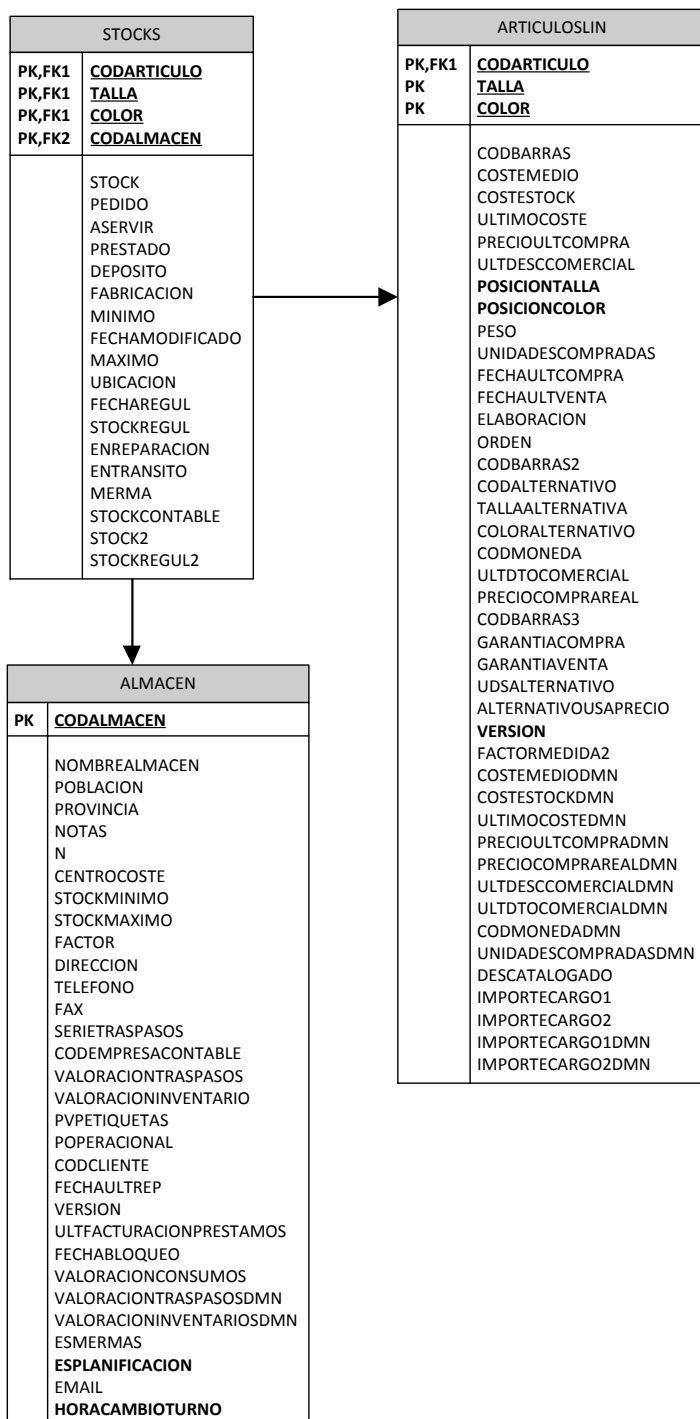
La taula principal d'articles és la **ARTICLES**, de la qual depèn una relació de composició amb **ARTICULOSLIN** (on per cada article s'especifiquen les diferents combinacions de talles i colors). A la vegada la taula **ARTICLES** conté atributs que es relacionen com a claus foranies amb altres taules del sistema:

- **DEPARTAMENTO, SECCIONES, FAMILIAS, SUBFAMILIES**: Taules utilitzades per a la classificació categòrica de l'article i que s'utilitzen per definir la classificació de tots els articles de la base de dades. La relació però no té perquè ser completa en els 4 nivells, o sigui que poden existir articles associats a un departament, però no a cap secció, família o subfamília, o poden existir articles amb tots els nivells de categorització assignats.
- **IMPUESTOS**: Per definir el tipus d'impost associat a un article.
- **TALLAS**: Es referencia aquesta taula per assignar el codi de tallatge associat a l'article.
- **MARCA**: Per assignar la marca de l'article a una de les existents al sistema.
- **NORMAS, TACONES, TEMPORADAS i COMPOSICIONES**: Utilitzades per categoritzar els detalls de constitució de l'article, la seva funcionalitat o composició.



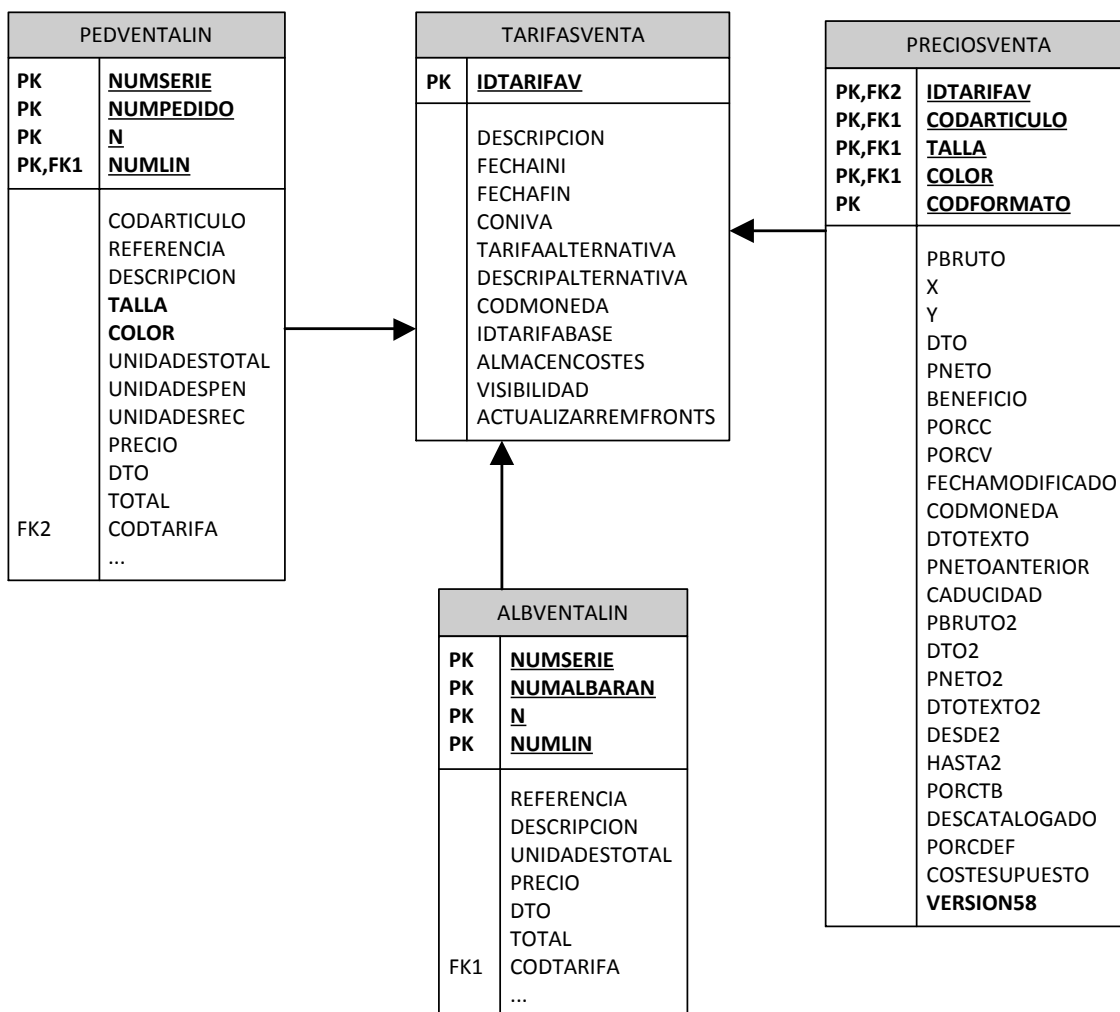
## Estocs

Aquest bloc del model es basa en l'existència d'una taula **STOCKS**, que es relaciona amb **ARTICULOSLIN** i **ALMACEN**. Això permet definir, per a cada combinació d'article (codi, talla i color) i cada magatzem, les seves disponibilitats i estats d'estoc.



## Preus i tarifes de venta

La taula principal de referència és **TARIFASVENTA**, a la qual els diferents documents de venta, ja siguin comandes o albarans (a través de **PEDVENTALIN** o **ALBVENTALIN**) es relacionen per assignar una tarifa associada a la venta (Utilitzant el camp **CODTARIFA**). A la vegada tots els preus existents dels articles es relacionen amb la taula **TARIFASVENTA** com anota el bloc del model que mostrem a continuació. Això permet que podem establir tots atributs de **PRECIOSVENTA** per cada combinació existent d'article (codi, talla i color) i tarifa (a través del seu identificador enter).



### 6.1.2 Model de dades de la web comercial

La base de dades de la web corporativa s'utilitza bàsicament per dos objectius:

- Permetre una representació intermèdia de les entitats que es requereixen per la comercialització: Articles, preus, estocs,... I no dependre d'una connexió real amb el sistema. Això implica un procés d'actualització diari d'aquestes dades, que no ens afecta al nostre sistema ja que la gestió de dit procés està a càrrec de l'empresa desenvolupadora de la web.
- Permetre emmagatzemar la informació necessària pel correcte funcionament de l'aplicació web comercial (usuaris, imatges, seccions, cistella de la compra, ...)

No obstant, existeixen alguns processos que en un futur no molt llunyà l'empresa voldrà que es gestionin des del mateix SetelSys. Justament per això, i tenint en compte que ja hi ha un procés que requereix de la connexió amb la base de dades de la web, aprofitem aquest apartat per veure parts del model de dades de la web que ens farà falta conèixer i controlar per futures ampliacions de funcionalitats.

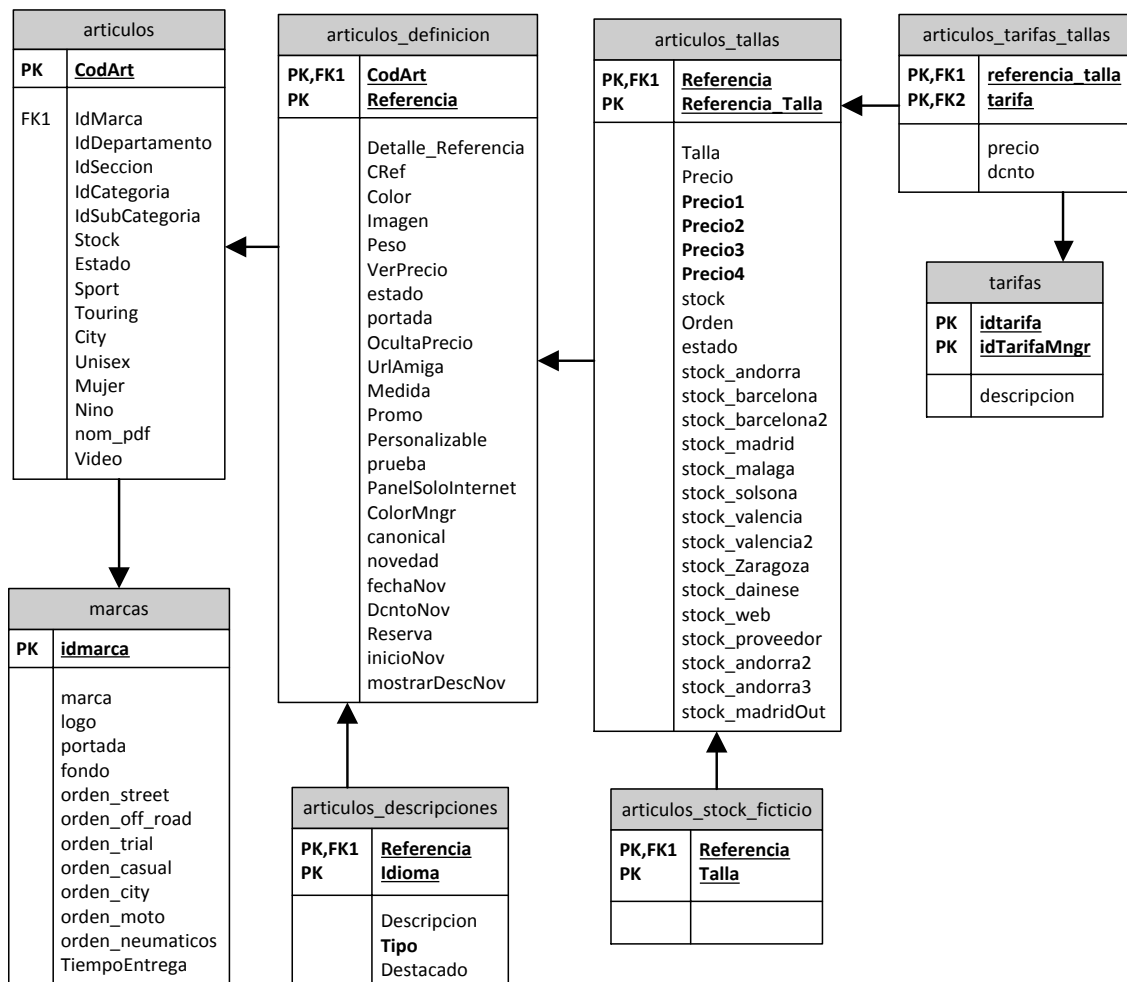
Així com l'ERP principal disposa d'una base de dades semànticament més ben construïda i en una forma correcta, no podem dir exactament el mateix de la pàgina web, en la que detectem certes irregularitats i un estat no precisament normalitzat.

## Articles i preus associats

La taula principal d'aquest bloc és la de *articulos*, on s'emmagatzema la informació bàsica i categòrica de l'article (marca, departament, secció, família, etc.). És important denotar que en aquesta base de dades, un article

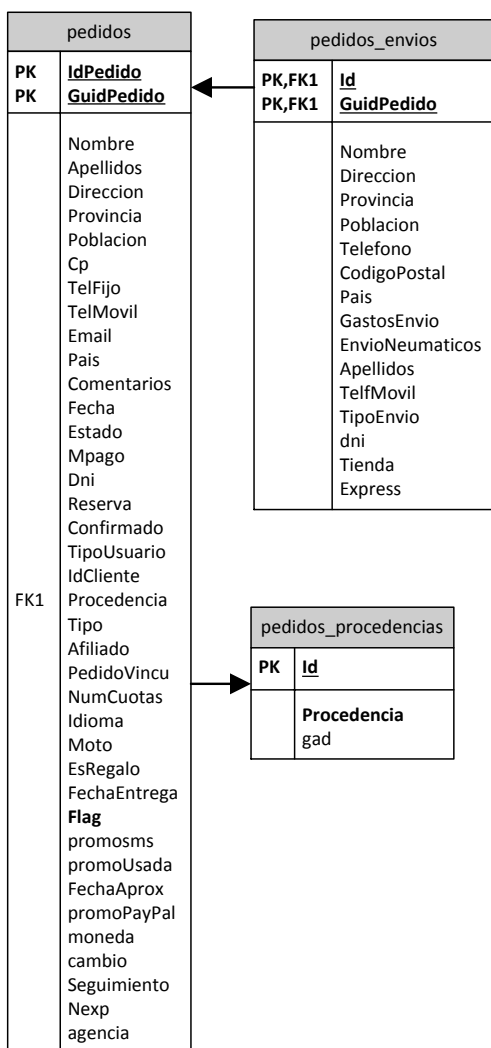
La taula *articulos\_definicion* es relaciona amb *articulos*, bàsicament per ampliar la informació associada a l'article de forma general (referència dins l'ERP, pes, estat, si és o no novetat,...) i la taula *marcas* relacionada a través de *idMarca*. D'aquesta taula *articulos\_definicion* en depenen.

- *articulos\_tallas*: On es defineix la descomposició d'un article en les seves diferents talles. Amb la qual es relaciona *articulos\_tarifas\_tallas* i *tarifas* (que contenen la informació dels diferents preus assignat a l'article en les diferents tarifes).
- *articulos\_descripciones*: On s'assigna la descripció de cada article, en cada un dels possibles idiomes existents en la web.



## Comandes generades a la pàgina web

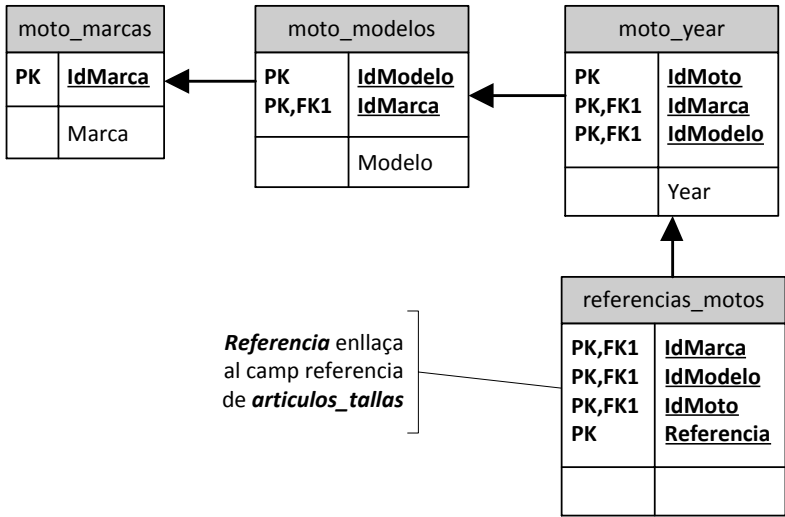
Les comandes que es generen dins la pàgina web s'emmagatzemen a la taula **pedidos**, a la qual s'hi relaciona **pedidos\_envios**, amb la informació de l'enviament de la comanda (dades d'enviament, tipus d'enviament, magatzem origen de l'enviament,...). També la taula **pedidos** es relaciona amb **pedidos\_procedencias** per a controlar l'origen de la pàgina web o webs que l'enllacen, des d'on s'ha iniciat la comanda.





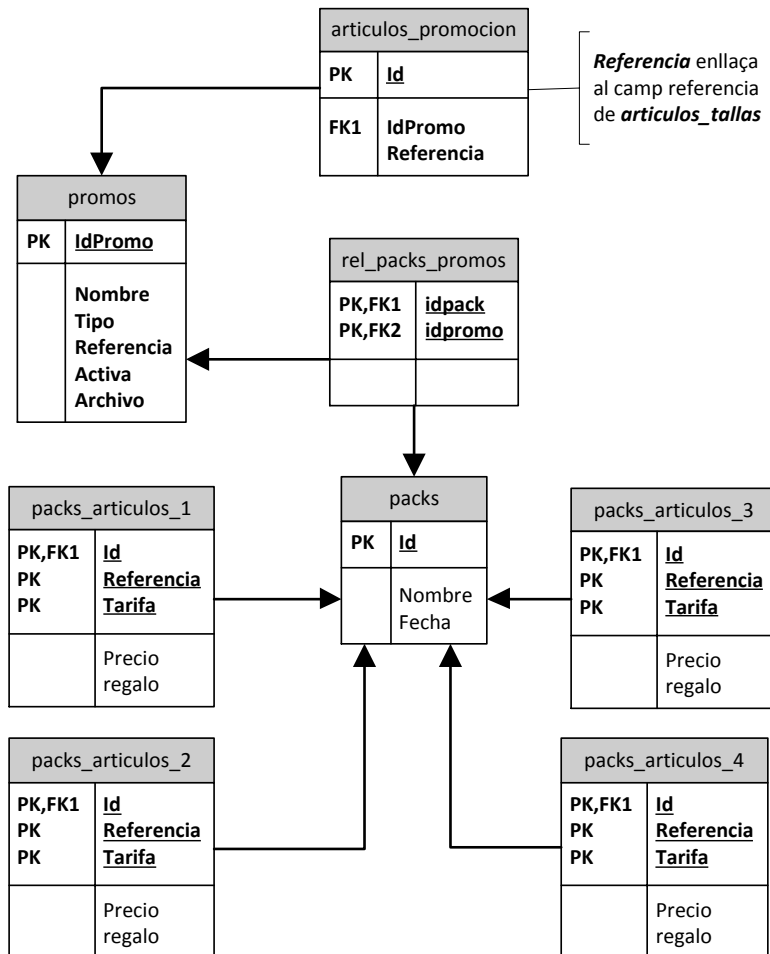
Gestió de motocicletes i articles associats

Des de la web es gestionen els articles associats als diferents models de motocicletes, aquestes es distribueixen en una cadena de relacions d'agregació, de forma que tenim la taula principal ***moto\_marcas*** que es compon de ***moto\_modelos*** i aquesta a la vegada de ***moto\_year***. Existeix una taula, ***referencias\_motos***, on s'associen els articles a les motocicletes existents a ***moto\_year*** i a la vegada es relaciona amb la taula ***articulos\_tallas***:



## Gestió de packs

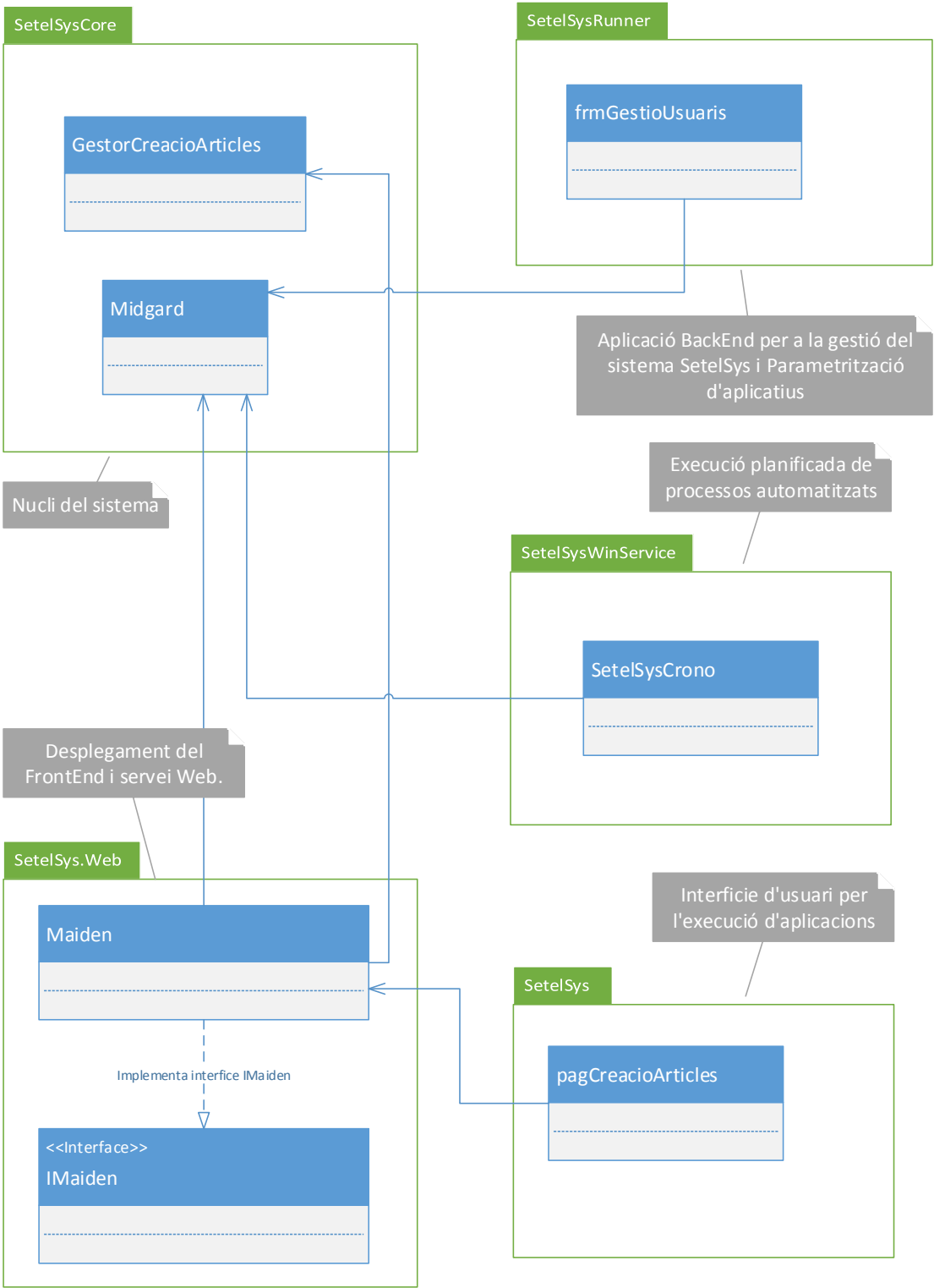
En la promoció de la venda d'articles a la pàgina web, aquests es registren a la taula de **packs** per permetre publicar ofertes especials. En concret de la taula **packs** es relacionen quatre taules amb combinacions diferents d'articles per aquell pack (**packs\_articulos\_x**). A la vegada existeix una taula de control **rel\_packs\_promos** que relaciona els diferents **packs** existents amb promocions representades a través de la taula **promos** on s'hi relacionen els articles que la conformen a través de **articulos\_promocion**.



# 6.2 Especificació de l'arquitectura del sistema SetelSys

## 6.2.1 Diagrama de components i dependències del sistema

Mostrem a continuació un diagrama del disseny dels diferents components del sistema i les dependències existents entre ells. Posteriorment fem una descripció dels diferents components.



## SetelSysCore

Nucli del sistema encarregat de gestionar tota l'operativa funcional amb:

- Connexió física amb les bases de dades. (A través de la classe **SQLServer**)
- Representació i gestió de l'operativa bàsica CRUD associada a les classes representatives de les entitats existents en el sistema ERP existent i la web comercial vinculada. (**WebDataCom** i **ERPDataCom**)
- Representació i gestió de l'operativa bàsica CRUD associada a les classes del sistema SetelSys (**SetelSysDataCom**).
- Gestió d'errors del sistema i missatges textuais associats a dits errors, així com la gestió de generació d'enviaments de correus i missatgeria (**Mercury**).
- Implementació dels diferents gestors d'aplicacions, encarregats d'executar les tasques especialitzades demanades per cada aplicació.
- Implementació d'un controlador especialitzat en servir les connexions i accés als diferents components del sistema (**Midgard**).

## SetelSysRunner

Interfície realitzada amb WinForms, encarregada d'implementar l'operativa corresponent a:

- Gestionar els usuaris, botigues, permisos, aplicacions, ...
- Gestionar les parametritzacions especialitzades requerides per algunes de les aplicacions o funcionalitats desitjades pel client.
- Gestionar les aplicacions automatitzades actualment vigents en el sistema així com la seva parametrització i patrons d'execució.
- Visualitzar el registre d'activitat de les aplicacions automatitzades.
- Eines per l'agilització del manteniment.
- Incorpora totes les classes referents a formularis d'interacció gràfica amb l'usuari administrador del sistema **SetelSys**.

## SetelSysWinService

Servei de Windows encarregat de l'execució en segon pla de totes les aplicacions automatitzades segons el patró que tenen establert. Utilitza la classe **SetelSysCrono** per a formalitzar la interacció amb el nucli (**SetelSysCore**) i obtenir la informació de les aplicacions vigents, paràmetres i temporització. L'interval de verificació d'execucions és cada 1 minut i s'inicia de forma automàtica a l'iniciar el sistema (independentment de l'inici de sessió de cap usuari), utilitzant la compta d'usuari local del sistema.

## SetelSys.Web

Component encarregat de carregar la interfície gràfica d'usuari elaborada amb Silverlight i que utilitzaran els usuaris consumidors del sistema, a diferència dels administradors que tindran les tasques centralitzades a través del **SetelSysRunner**.

Dit component allotja també la definició del servei web **Maiden**, encarregat de connectar-se amb el **SetelSysCore** per efectuar les operacions necessàries per les diferents aplicacions utilitzant classes heretades de **GestorAplicacio**.

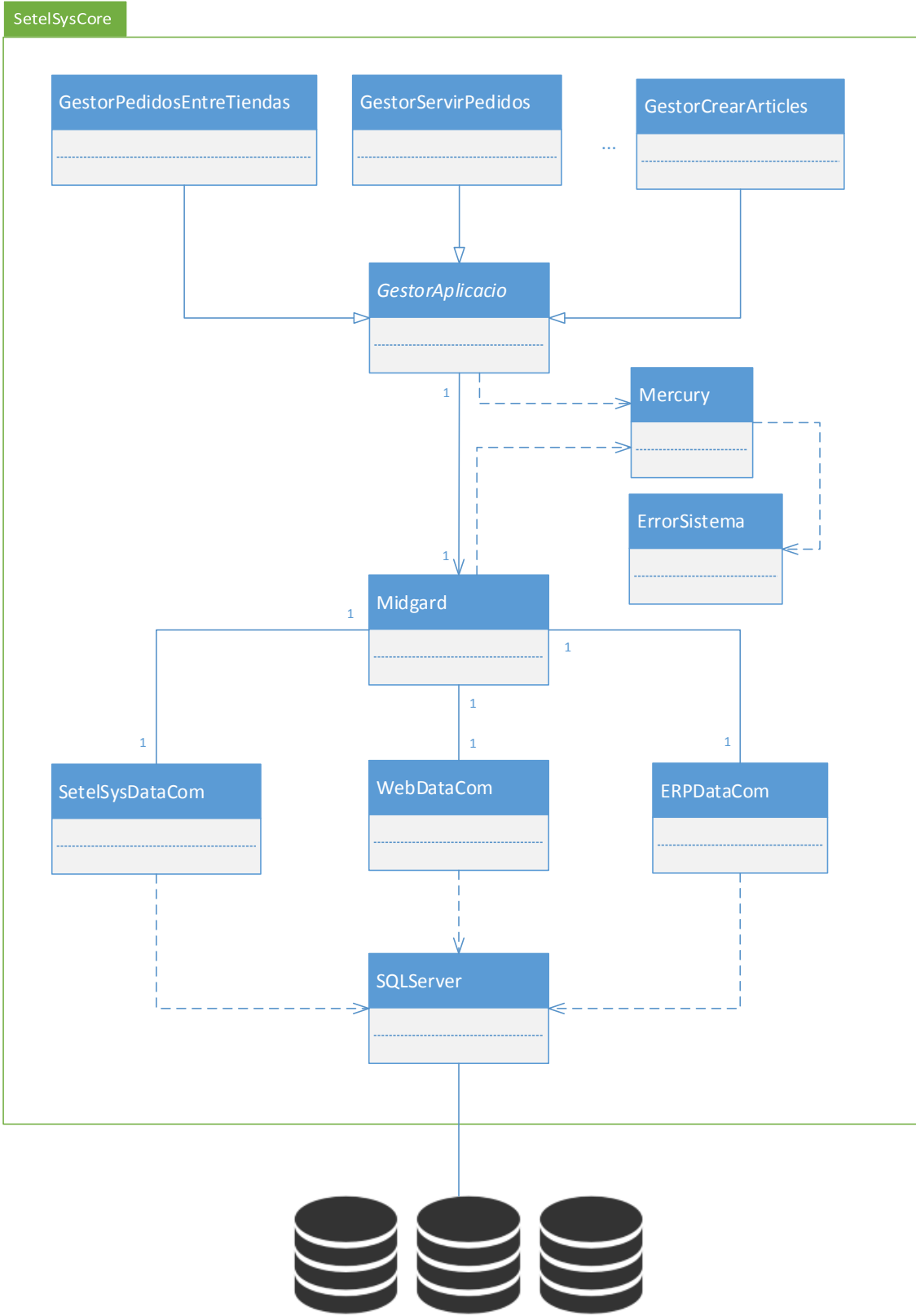
## SetelSys

Interfície d'usuari elaborada amb Silverlight, encarregada de mostrar les pantalles d'interacció amb l'usuari consumidor. El component s'estructura en pàgines i finestres de Silverlight a través d'un menú principal generat dinàmicament en base dels permisos de l'usuari que es connecta pel que fa a l'execució de les aplicacions disponibles.

Tota interacció de dades entre la interfície i el nucli principal s'efectuen a través de la crida del servei **Maiden** allotjat en el component anteriorment descrit **SetelSys.Web**.

6.2.2 Diagrama de classes del nucli del sistema (SetelSysCore)

Com es podrà observar en el següent diagrama, el nucli s'estructura a través de les classes següents:



Les diferents classes que hereten de **GestorAplicació**, s'utilitzen per a mapejar les funcionalitats assignades a les diferents aplicacions del sistema. Disposen d'accés a la classe **Midgard**, encarregada de gestionar els recursos del sistema i la qual té una instància de les diferents classes de funcionalitat bàsica encarregats de la comunicació amb les dades finals a través de la classe **SQLServer**. Existeix un comunicador de dades per cada element existent al sistema i pel propi sistema SetelSys dissenyat:

- **ERPDataCom**: Encarregat de gestionar la funcionalitat d'accés i representació de les diferents entitats de l'ERP.
- **WebDataCom**: Encarregat de gestionar la funcionalitat d'accés i representació de les diferents entitats de la web comercial.
- **SetelSysDataCom**: Encarregat de gestionar la funcionalitat del sistema de control annex necessari per un correcte funcionament i compliment de requisits encarregats en l'anàlisi de requeriments.

Finalment cal destacar la dependència existent entre les classes GestorAplicacio i Midgard, amb la classe Mercury. L'objectiu de la qual ja hem esmentat anteriorment i es pot resumir en la gestió dels errors detectats en el sistema i de la gestió multilingüe futurament necessària, així com la gestió d'enviaments de correu i missatgeria en general.

### 6.3 Especificació del gestor del sistema (SetelSysDataCom)

Aquest component, com hem dit abans s'encarregarà de la gestió de les necessitats i infraestructura pel correcte funcionament del sistema de suport per l'execució d'aplicacions, gestor de permisos i en general la gestió de les diferents entitats i classes necessàries.

A continuació mostrem un anàlisi dels casos d'ús detectats per l'elaboració del disseny del sistema:

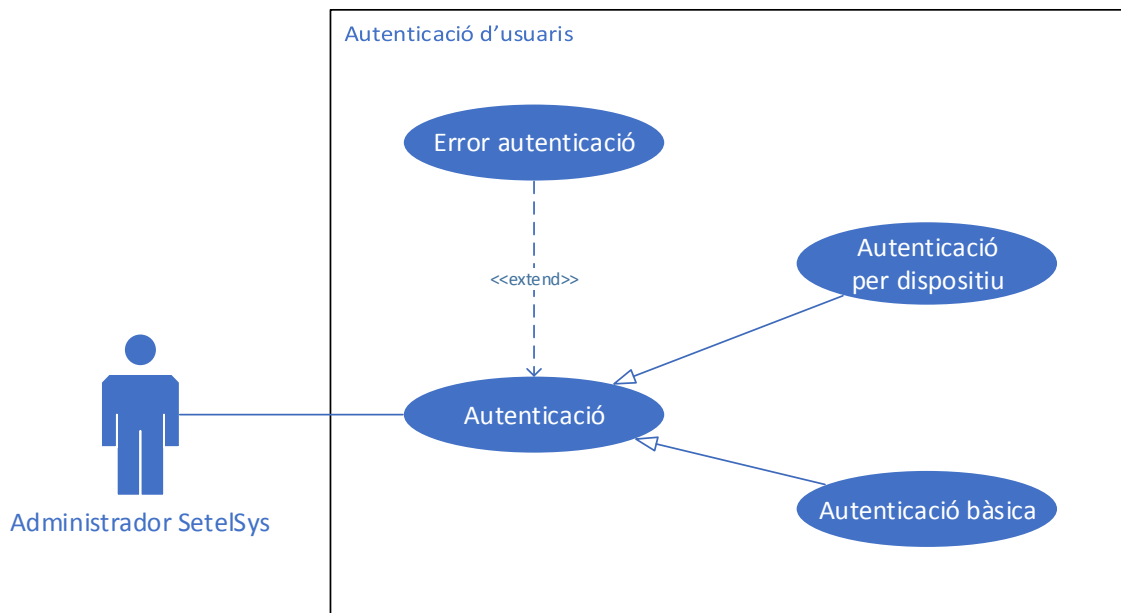
#### 6.3.1 Casos d'ús

##### Autenticació d'usuaris

*Qualsevol usuari, tant a nivell administratiu com estàndard caldrà que es validi en el sistema per acreditar-se com a autoritzat i poder disposar dels aplicatius disponibles pel seu usuari. En concret es poden donar 2 possibles vies d'autenticació:*

- *Autenticació bàsica: Basada únicament en un nom d'usuari i contrasenya.*
- *Autenticació per dispositiu: A part de l'usuari i contrasenya, caldrà validar si l'origen de la connexió està autoritzat.*

*El tipus de validació de l'usuari aplicada serà en base de la configuració general del sistema. En cas de produir-se un error d'autenticació caldrà notificar-ho degudament a l'usuari involucrat.*



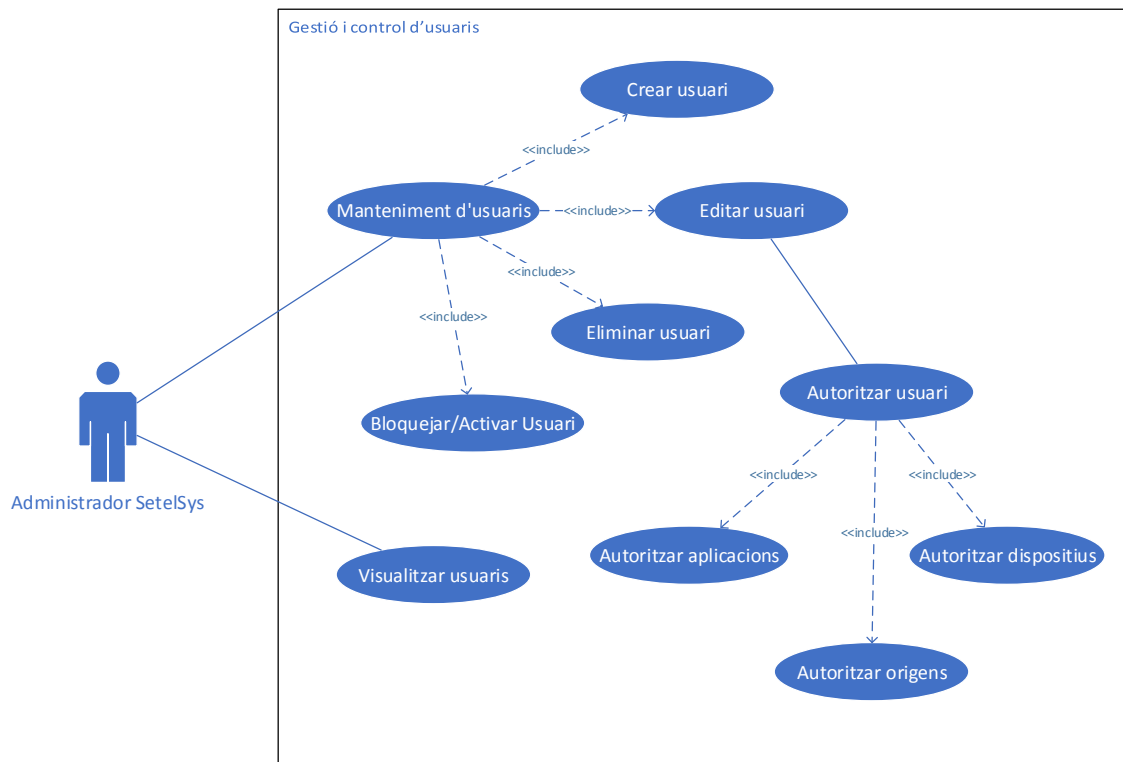
En els següents casos d'ús, per facilitar la lectura, suposarem que tota interacció es realitza sota el corresponent mecanisme d'autenticació.



## Gestió d'usuaris

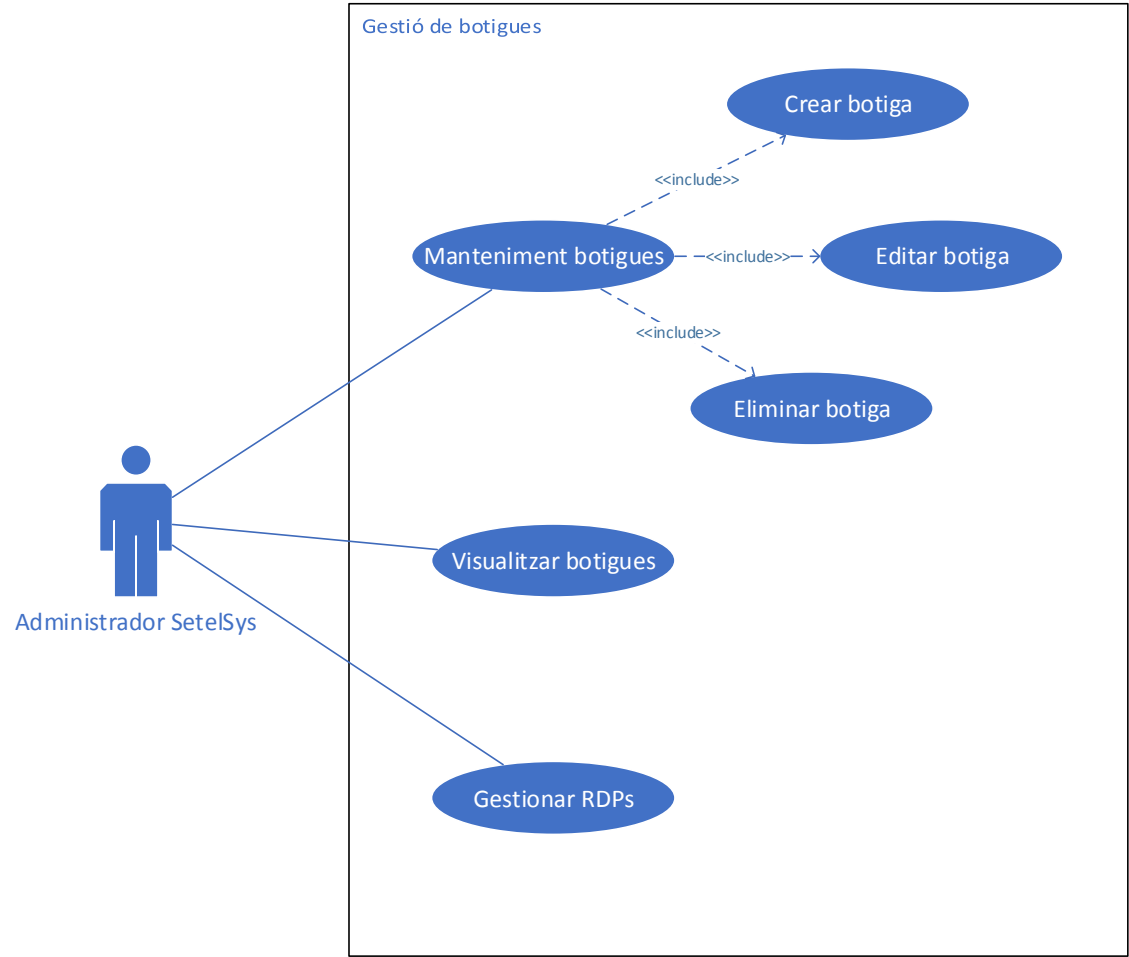
Cal dotar al sistema de la capacitat de gestionar els usuaris que es podran acreditar per accedir al sistema. Caldrà poder llistar i aplicar el manteniment sobre qualsevol usuari, i també poder-li assignar la corresponent autorització de connexió en base de:

- Aplicacions autoritzades
- Dispositius d'origen de la connexió
- Botiga o central logística des de d'on s'inicia la connexió.



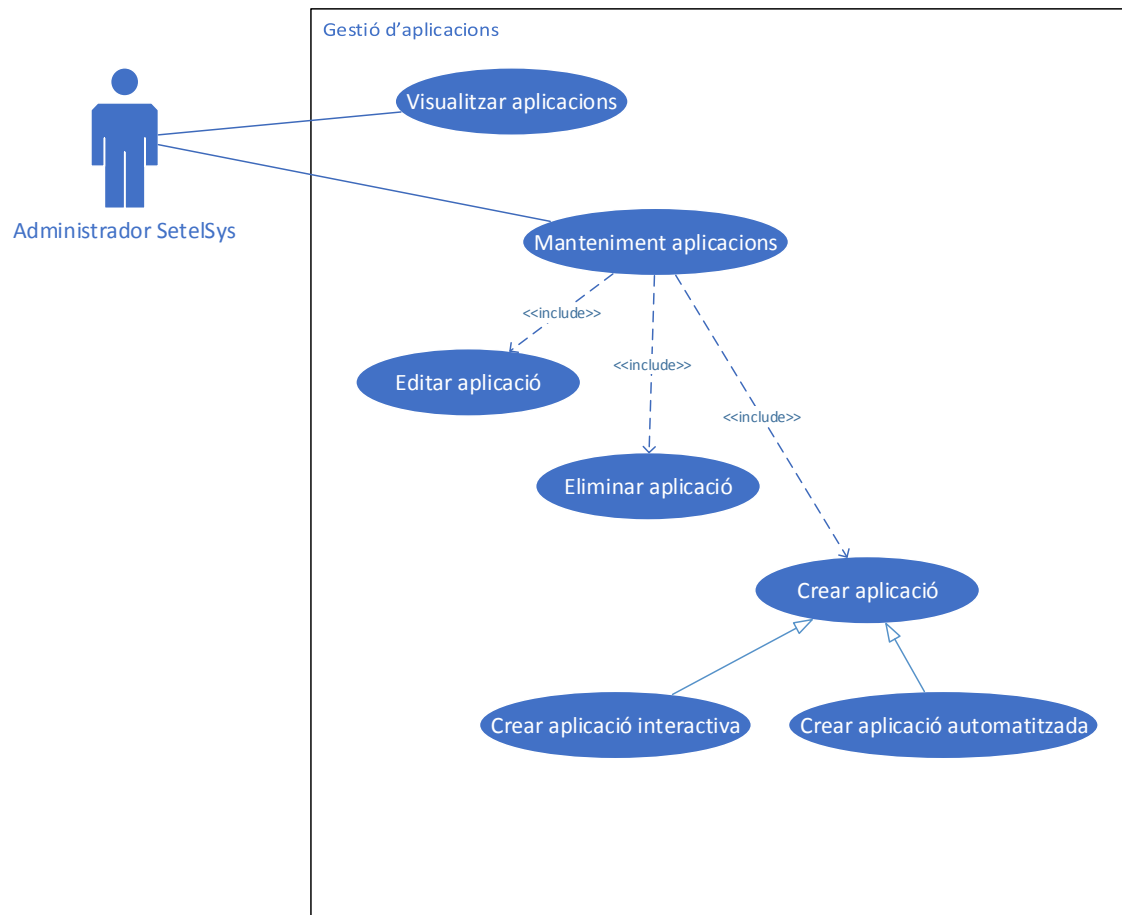
**Gestió de botigues**

Caldrà poder llistar, gestionar i mantenir les diferents botigues registrades en el sistema i que s'utilitzaran com origen de connexió i validació de l'acreditació. També caldrà dotar al sistema dels recursos necessaris per poder assignar RDPs (Remote Desktop Protocol) a les botigues en qüestió per que el usuari puguin descarregar-se dit fitxer generat de forma dinàmica des de l'aplicació.



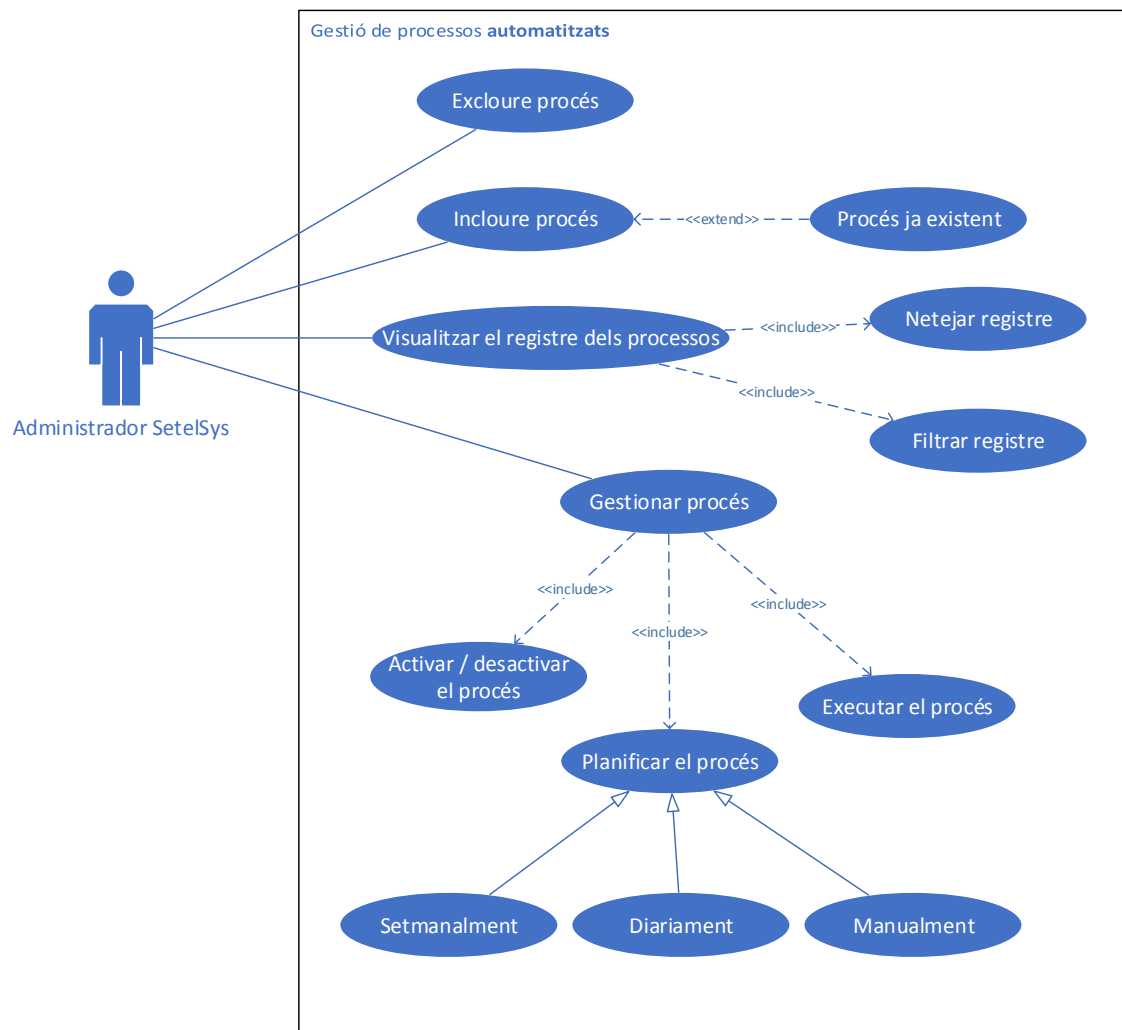
## Gestió d'aplicacions

Caldrà poder llistar les diferents aplicacions existents al sistema, així com gestionar-les i poder-ne crear de noves, ja siguin aplicacions interactives o bé aplicacions automatitzades (sense interacció amb l'usuari ja que tot el procediment està parametritzat per la definició dels seus objectius i es pot executar sense parametrització de l'usuari)

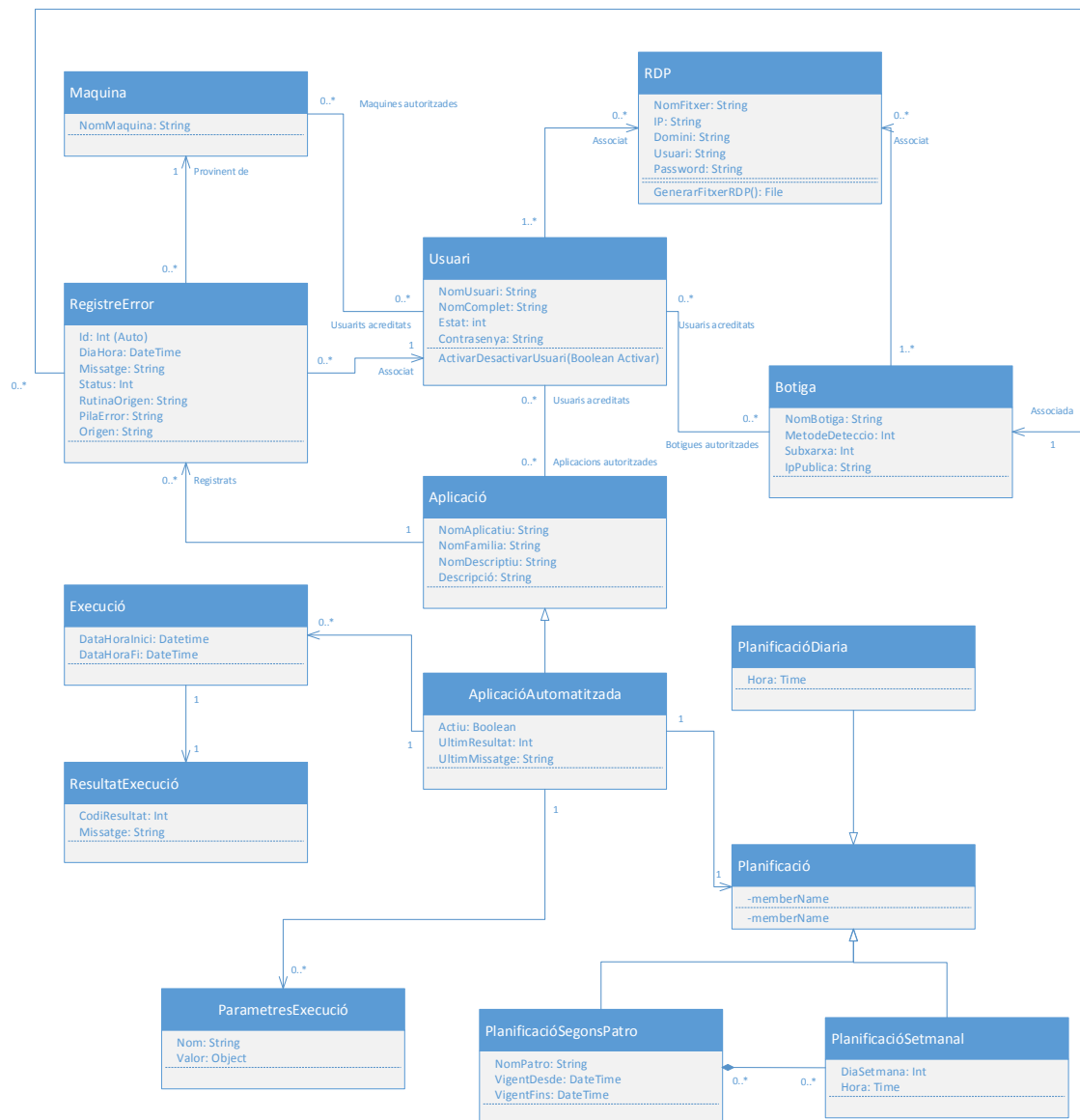


## Gestió de processos automatitzats

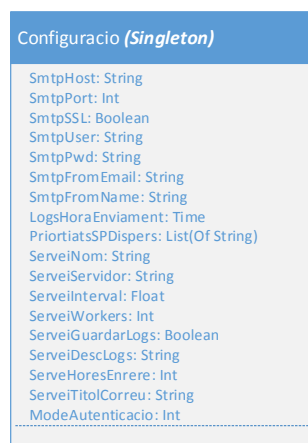
Caldrà que l'usuari pugui gestionar un procés en tota la seva complexitat, podent modificar el patró d'execució, activar-lo o desactivar-lo i executar-lo sota demanda. D'altra banda també caldrà poder incloure i excloure el procés en la llista de processos vigents en la planificació global. Finalment s'haurà de poder visualitzar el registre de processos, podent-lo netejar i filtrar a través d'un interval de dates a escollir.



### 6.3.2 Diagrama asociativu de classes

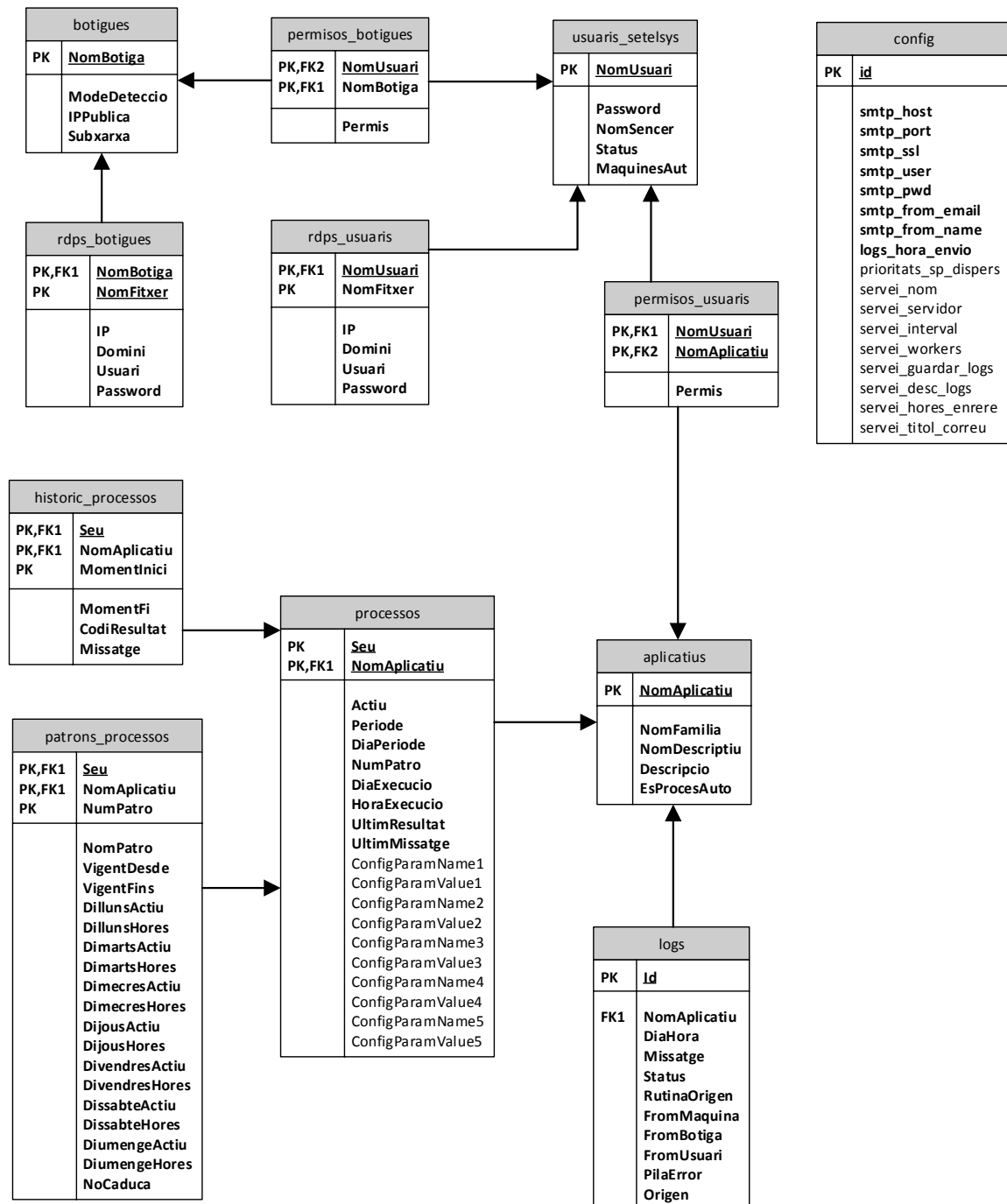


Existirà també una classe aïllada del diagrama, de tipus **Singleton** encarregada d'emmagatzemar paràmetres de configuració del sistema, anomenada **Configuració**:



## 6.4 Disseny de la base de dades de SetelSys

Mostrem tot seguit el model relacional resultant de l'especificació anterior, del qual tot seguit en mostrarem una explicació dels diferents grups de taules creats i la seva utilitat i funció com a membres del model de dades:



### *Gestió d'usuaris*

A través de la taula “*usuaris\_setelsys*” gestionem els diferents usuaris presents en el sistema, amb el seu nom, contrasenya i nom d'usuaris.

### *Gestió dels permisos dels usuaris*

Les màquines autoritzades decidim d'implementar-les com un atribut de tipus llista dins la taula d'usuaris. En canvi els permisos referents als aplicatius i les botigues, es guarden respectivament a “*permisos\_usuaris*” que formalitza l'implementació d'una relació N:N entre els usuaris els aplicatius, i a “*permisos\_botigues*” que a la seva part formalitza la relació N:N entre els usuaris i les botigues. També podem veure la relació existent entre la taula “*rdps\_usuaris*” amb la taula d'usuaris, que assigna els diferents *rdps* als corresponents usuaris.

### *Assignació de RDPs:*

Hem subdividit l'entitat RDP en dos taules “*rdps\_usuaris*” i “*rdps\_botigues*” que es relacionen amb usuaris i botigues respectivament. Hem detectat que ens certs casos es possible que en un futur es requereixi una especialització diferent per cadascuna, deixant així la porta oberta futures possibles modificacions o ampliacions de forma fàcil i trencant possibles dependències.

### *Botigues:*

Dins la taula “*botigues*” emmagatzemem les diferents botigues disponibles al sistema, juntament amb el seu nom identificador i el mètode que cal utilitzar per l'auto detecció de l'origen (utilitzant els atributs *IPPublica* i *Subxarxa*).

### *Aplicatius i processos:*

Dins la taula “*aplicatius*” s'emmagatzemen els diferents tipus d'aplicacions existents al sistema, així com la seva categorització (*NomFamilia*) i si es pot executar de forma automatitzada. En el cas de poder-se automatitzar es relaciona amb la taula “*processos*” on s'emmagatzema la informació del patró d'execució juntament amb la relació existent amb la taula “*patrons\_processos*” així com els paràmetres de configuració. Ens hem adaptat al màxim a la realitat de les necessitats per evitar costos de taules innecessàries, limitant el número de paràmetres a configurar en 5, amb la qual cosa podem definir-los dins la taula “*processos*” i no calent realitzar una nova taula i noves relacions.

### *Registre d'errors i històric de processos*

En la taula “*logs*”, guardem la informació de qualsevol error registrat al sistema amb tota la informació que el defineix així com la seva traçabilitat en la pila de l'error. D'altra banda a la taula “*històric\_processos*” guardem tota la informació referent a les diferents execucions realitzades dels diferents processos automatitzats vigents al sistema.

## 7. Desenvolupament

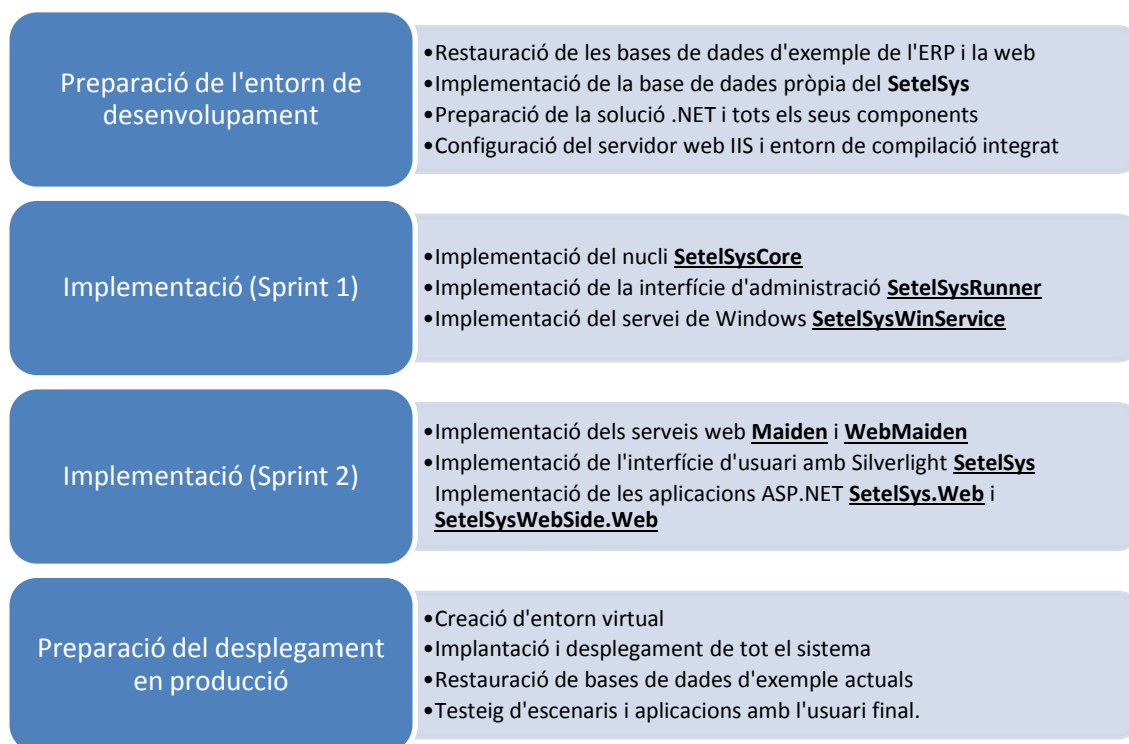
### 7.1 Metodologia utilitzada per al desenvolupament

Per a un correcte desenvolupament hem dut a terme les següents tasques inicials

- Elecció de les eines utilitzades: Visual Studio, SQL Server, SQL Studio Management, Microsoft Visio, Silverlight SDK v4, Blend Expression, Microsoft IIS i llibreries addicionals.
- Estructuració física dels arxius en l'entorn de desenvolupament, per permetre un desplegament ràpid en l'entorn de desenvolupament i poder-lo reproduir en producció.
- Automatitzacions en el procés de compilació i implantació de la solució per agilitzar les proves bàsiques en l'entorn de desenvolupament.
- Establir les tècniques de documentació a utilitzar, tant en el codi com en la definició de classes, mètodes, funcions,...
- Enumerar i establir les premisses i consideracions a tenir en compte durant el desenvolupament: Evitar la redundància de codi, reaprofitar codi existent, documentar, aplicar les nomenclatures escollides,...



## 7.2 Cronologia en el desenvolupament



## 7.3 Estructura de la solució .NET

La tipologia de cada un dels components que conformen la solució és la següent:

- **SetelSys**: Aplicació Silverlight<sup>13</sup>, hostatjada i iniciada a través del component SetelSys.Web i encarregada de subministrar una interfície per interaccionar amb l'usuari en l'execució dels diferents aplicatius. Es connecta a través del servei web Maiden, que a la vegada ho fa amb el nucli del sistema.
- **SetelSys.Web**: Aplicació ASP.NET que inicialitza i carrega l'arxiu .XAP<sup>14</sup> generat en la compilació de SetelSys, i a la vegada allotja el servei web Maiden, basat en WCF<sup>15</sup>, per la interacció del component SetelSys amb els objectes del sistema.
- **SetelSysCore**: Llibreria de Windows (.DLL) encarregada d'implementar tot el nucli neuràlgic del sistema, representant els objectes i capes necessàries per suplir tota la lògica de negoci necessària.
- **SetelSysRunner**: Aplicació basada en WinForms<sup>16</sup> per actuar com a interfície d'usuari d'administració i parametrització del sistema, utilitzant SetelSysCore per la connexió amb la lògica de negoci i la gestió dels objectes de l'aplicació del sistema principal.

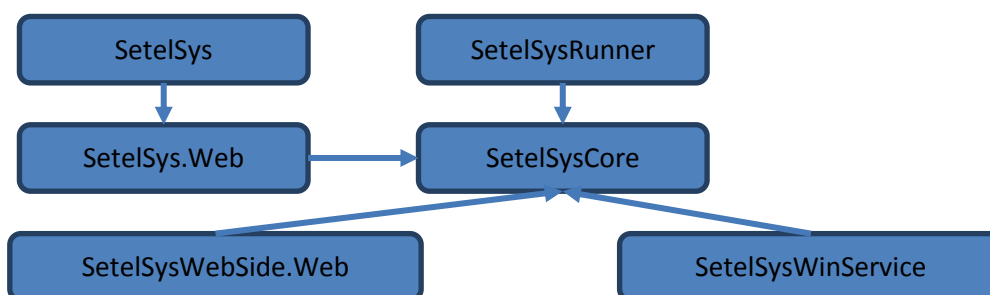
<sup>13</sup> Que és Silverlight? – [Microsoft.com](http://Microsoft.com)

<sup>14</sup> Definició sobre els arxius .XAP - [Wikipedia](http://Wikipedia)

<sup>15</sup> Hosting and consuming WCF Services – [Microsoft MSDN](http://Microsoft MSDN)

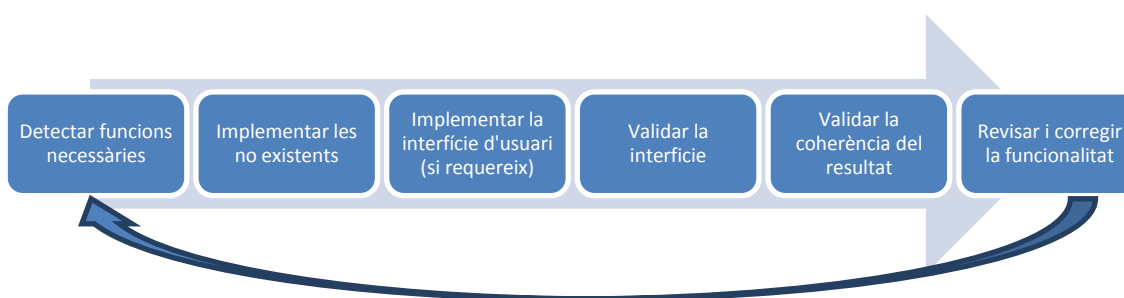
- **SetelSysWebSide.Web:** Aplicació ASP.NET<sup>17</sup> encarregada d'hostatjar el servei web necessari en l'allotjament de la web comercial per poder accedir a les dades i objectes representatius de la web.
- **SetelSysWinService:** Servei de Windows (Windows Service<sup>18</sup>) que implementa el motor de gestió d'execució de processos automatitzats.

En el següent diagrama mostrem les referències existents en la solució, entre els diferents projectes dels components existents, i responen a l'especificació de l'arquitectura de la solució:



## 7.4 Cicle de vida en la implementació de funcionalitats

Durant el desenvolupament de noves funcionalitats, s'ha seguit un procés encadenat per complir amb les premisses desitjades en el sistema resultant. En concret en tota nova funcionalitat implementada s'ha prosseguit seguint el següent diagrama representatiu:



<sup>16</sup> Windows Forms – [Microsoft MSDN](#)

<sup>17</sup> ASP.NET Web Forms – [ASP.Net](#)

<sup>18</sup> Introduction to Windows Service Application – [Microsoft MSDN](#)

## 7.5 Consideracions preses en el disseny de les UI i interacció amb l'usuari

En la implementació de les interfícies d'usuari, ja sigui amb WinForms (SetelSysRunner) o Silverlight (SetelSys), s'han pres un seguit de consideracions pel que fa a les interaccions que es generen amb l'usuari. Les enumerem tot seguit:

- Notificacions adequades i estandarditzades a través de finestres emergents, per comunicar notificacions, avisos, errors o interactuar en la presa de dades.
- Establir els mecanismes d'ajuda a l'usuari en els formularis o pàgines, a través d'ajudes contextuais i finestres d'ajuda, per donar eines a l'usuari pel coneixement de les aplicacions.
- Utilització sistemàtica de controls similars o idèntics per a funcionalitats semànticament semblants per no confondre l'usuari en l'ús dels diferents controls.
- Utilització sistemàtica d'imatges identificadores per notificar l'usuari sempre amb les mateixes icones lligades a valors semàntics associats.

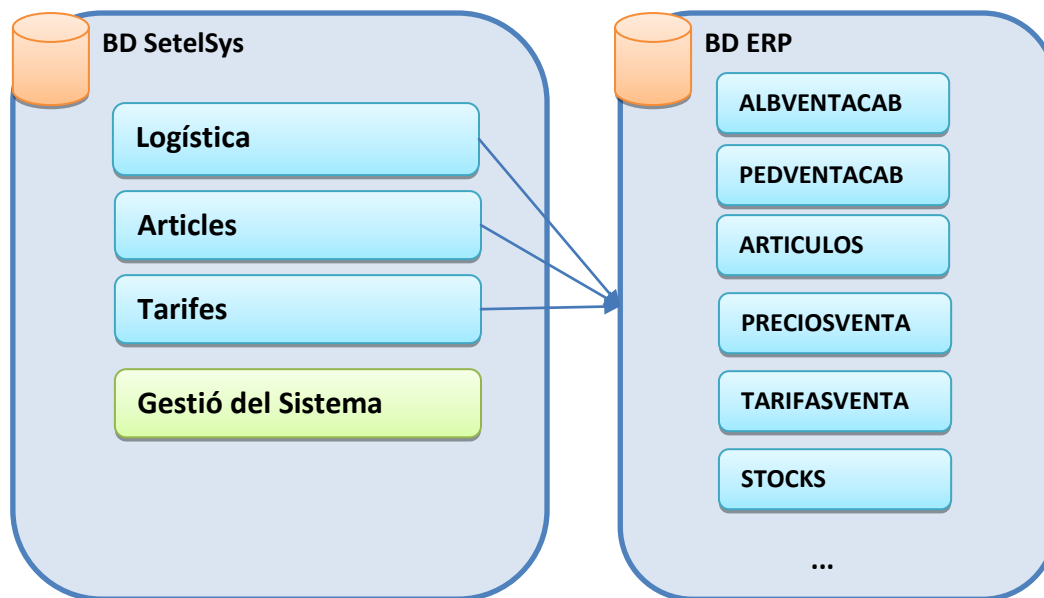
## 7.6 Implementació de la base de dades principal de SetelSys

La base de dades de SetelSys implementarà les entitats i relacions necessàries per:

- Gestionar la infraestructura del sistema SetelSys.
- Gestionar la interacció del sistema amb altre subsistemes: ERP + Web

Això genera una casuística especial, ja que si bé per les taules referents al sistema SetelSys podem gestionar-les totes dins la nostra base de dades a implementar, les taules relacionades amb l'ERP i la web comercial, **no poden plasmar les relacions existents al formar part de dos bases de dades independents lògica i físicament.**

Per aquest motiu, la base de dades de SetelSys implementa les entitats necessàries pel funcionament de les aplicacions dependents de l'ERP (les tasques a gestionar a la web comercial no requereixen de creació de taules auxiliars), sense existir una relació a nivell estructural de la base de dades, però sí semàntic, entre les entitats.



## 7.7 Implementació dels components del projecte

### 7.7.1 SetelSysCore

Al nucli del sistema, SetelSysCore, li correspondrà entre d'altres:

- Donar representació a les entitats de la capa de negoci de l'empresa, relacionada amb les bases de dades de l'ERP i la web comercial.
- Donar una representació i funcionalitat als objectes de la gestió del sistema.
- Dotar al sistema de les eines de gestió d'errors i servei de missatgeria.
- Registrar l'activitat i les connexions establertes autenticant-les correctament.
- ...

### 7.7.2 SetelSysRunner

SetelSysRunner ha estat desenvolupat a través de WinForms<sup>19</sup> (formularis per a Windows), com a eina de control administratiu del sistema i de parametrització de les diferents aplicacions existents en la interfície d'usuari SetelSys feta amb Silverlight. Recordem breument, tot i haver-se anotat anteriorment, els objectius d'aquesta aplicació administrativa:

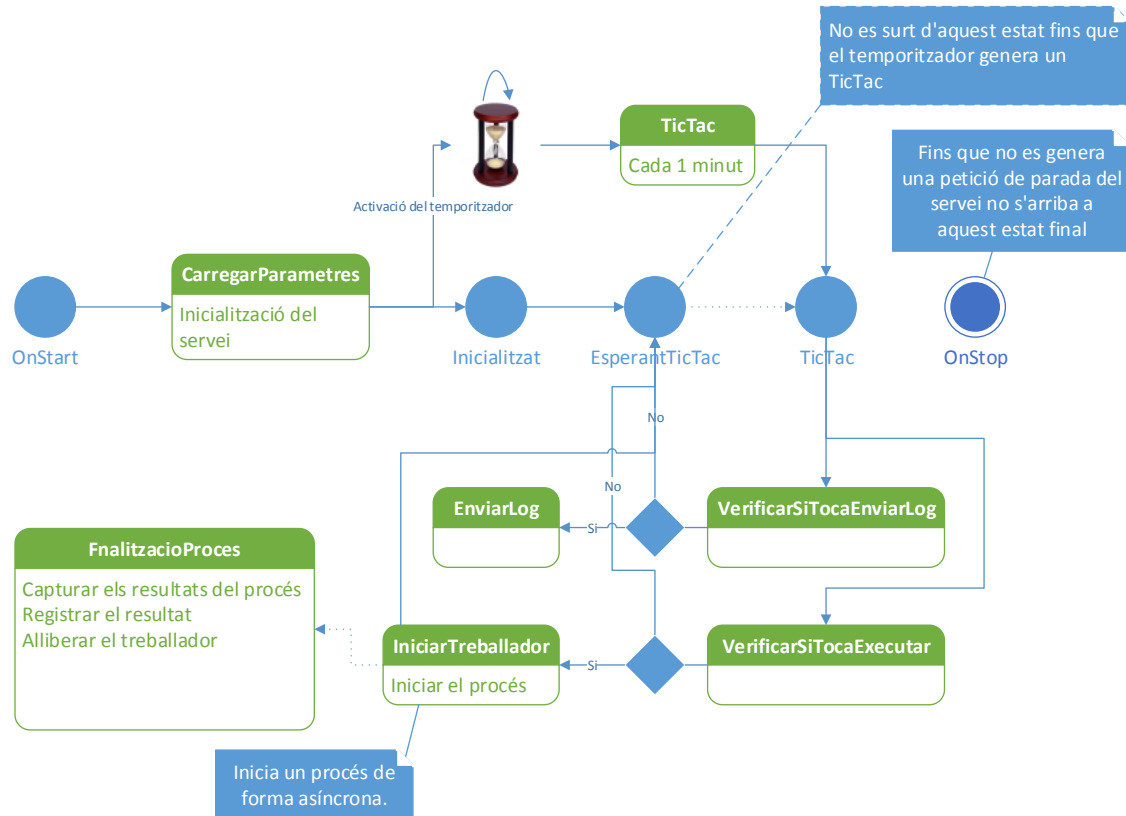
- Gestionar els **processos** automatitzats existents (afegir, editar, eliminar,...).
- Visualitzar i operar amb el **registre** d'activitat dels processos automatitzats.
- Gestionar les **entitats** del sistema SetelSys: Usuaris, permisos, botigues, aplicacions, notificacions,...
- **Parametritzar** processos complexos. Estem parlant en tot moment de parametritzacions que només cal poder editar puntualment per canviar paràmetres globals del funcionament de les diferents aplicacions.
- Dotar **d'eines** de consulta per als **desenvolupadors** a nivell intern i privat (activables només amb una combinació de tecles).

---

<sup>19</sup> Windows Forms – [Microsoft MSDN](#)

### 7.7.3 SetelSysWinService

Aquest component serà l'encarregat d'implementar un servei de Windows<sup>20</sup> per verificar de forma intermitent les necessitats d'execució d'algun dels processos automatitzats del sistema.



### 7.7.4 Maiden: Servei web

Dins del projecte **SetelSys.Web** hem definit el servei web Maiden, basat en l'arquitectura de serveis WCF de Microsoft.

Aquest servei serà l'encarregat de transmetre totes les peticions de l'aplicatiu Silverlight al nucli del sistema resident en el servidor d'aplicacions i dades, donant accés a tota la lògica de negoci necessària per mostrar a l'usuari final.

### 7.7.5 SetelSys: Aplicació web amb Silverlight.

Aquesta aplicació és l'encarregada d'implementar la interfície d'usuari a través d'una aplicació web encapsulada amb Silverlight.

<sup>20</sup> Create Windows Services – [Microsoft MSDN](#)

## 7.8 Implementació d'eines d'ajuda al desenvolupament i actualització a noves versions.

Per tal d'ajudar el procés de verificació dels entorns de desenvolupament i producció, així com pel desplegament de noves actualitzacions i versions al sistema en producció, hem elaborat un seguit d'eines per a facilitar-nos en aquestes tasques:

- **DBComparer:** Compara les definicions de dos bases de dades per tal de poder-ne detectar variacions.
- **DBTester:** Verifica la connectivitat amb les bases de dades del sistema (SetelSys, ERP i Web)
- **FTP Deployer:** Utilitza el servei de FTP com a intermediari pel desplegament de noves versions, copiant tots els arxius necessaris des de la màquina de desenvolupament cap al FTP i posteriorment del FTP en el servidor de producció, situant els arxius descarregats en les rutes especificades en la configuració de l'aplicació FTP Deployer.
- **Chrono Checker:** S'encarrega de monitoritzar l'estat d'un servei. En concret el volem utilitzar per a monitoritzar l'estat de SetelSysWinService, de forma que sempre tinguem un control visual de l'estat del servei.

## 7.9 Llibreries i components externs utilitzats

Per al desenvolupament de la solució s'han utilitzat tant llibreries pertanyents al marc de treball .NET Framework, com altres llibreries externes per a la implementació de funcionalitats no natives al .NET:

- Connexions via FTP
- Creació d'informes amb Crystal Reports.
- Creació i lectura d'arxius d'Excel o separats per comes (CSV) amb JetCell (Llicència comercial amb un cost de 175,00 \$)
- Alguns controls i eines específiques per utilitzar dins del desenvolupament amb Silverlight.

## 8. Testing

Tal i com s'ha comentat en diverses ocasions al llarg de la memòria del projecte, a grans trets el projecte s'ha realitzat de forma conjunta amb el meu company; no obstant, dues grans àrees s'han realitzat de forma separada com són el desenvolupament i les fases de proves, de testing, inclosa una automatització de les mateixes.

Tot i que evidentment un projecte dut a terme només per dues persones no requeriria d'una aproximació tant potent, s'ha optat per una solució basada com es veurà en TMap, ja que, a part dels coneixements sobre la metodologia de la que disposàvem, és una metodologia molt interessant de cara a veure el potencial real del testing estructurat del software.

Tot i que actualment el **Testing de Software** i la **Quality Assurance** són cada vegada més presents en el món del desenvolupament de software, també s'ha de tenir en compte que ens molts casos el coneixement i l'ús d'enfocaments estructurats és molt escàs, fins i tot en l'àmbit mateix les facultats d'informàtica.

El present apartat de Testing, dins de la memòria, s'ha enfocat amb la idea d'oferir una introducció al món del software de testing, amb la idea d'emmarcar correctament els passos que s'han seguit a continuació en aquesta part del projecte de final de carrera conjunt. Seguidament s'explica la metodologia seleccionada, i es comença a entrar en matèria concreta amb les diferents fases que s'han dut a terme. I finalment, es mostra el següent gran grup de tasques que s'han realitzat per al projecte, emmarcades dins de l'automatització de proves.



## 8.1 Introducció al Testing de Software

### 8.1.1 Per què Testing?

Segurament des de sempre la qualitat ha estat un dels paràmetres a tenir en compte en qualsevol tipus de projecte i negoci, també dins el món del software. Però actualment, amb la situació del mercat, les organitzacions i empreses cada vegada tenen més interès per aconseguir el màxim valor empresarial al seu programari. I consegüentment, una entrega eficient i ràpida del software, i una bona qualitat del mateix estan esdevenint cada vegada més importants, i el risc d'una baixa qualitat va creixent. El Testing o les Proves són una de les accions més importants que es poden incorporar en els procediments i metodologies d'una empresa per a controlar aquests riscos.

El Testing dona una informació vital de la idea de la qualitat que té un software o sistema, i per tant, dels riscos que pot implicar una possible baixa qualitat final. I la idea general, és que en base a aquest coneixement, la direcció dels projectes o empreses poden emprendre les accions necessàries sobre una base sòlida de coneixement.

Es poden enumerar molts beneficis que pot aportar la incorporació d'un procés de testing, però els principals que podríem mencionar són els següents <sup>21</sup>:

- La prevenció dels (alts) costos de reelaboració i les conseqüències dels danys que es puguin ocasionar a producció, gràcies als defectes que es troben durant les fases de proves i les correccions que es duen a terme dins el procés de desenvolupament del sistema. Alguns exemples de les conseqüències dels danys són: la pèrdua d'ingressos, la pèrdua de la reputació / marca, les reclamacions d'indemnització i les pèrdues de productivitat.
- La prevenció de danys als sistemes productius, gràcies al coneixement dels errors trobats durant les proves, i tot i no està resolt, estan marcats com a “errors coneguts”.
- Aconseguir / guanyar confiança en el producte
- Facilitar la bona gestió del projecte gràcies a oferir (tant a nivell de progrés com de qualitat) informació.

---

<sup>21</sup> Rex Black, keynote at Eurostar2002, Copenhagen

### 8.1.2 Breu introducció història al testing de software

L'any 1959 A.M. Turing va redactar "*Checking a Large Routine*"<sup>22</sup>, on ja es posava a discussió el fet d'utilitzar assercions per a realitzar proves de correcció; a grans trets però, les proves de software ja s'estaven posant de manifest abans, ja que sempre han anat de la mà de la pròpia evolució de la enginyeria del software, i si intentem organitzar-les segons tal i com s'orientaven les proves, en podríem dividir la seva història en aquests grans grups:

- Abans de 1956: Període orientat a debugging
- Entre 1957 i 1978. Període orientat a demostració.
- Entre 1979 i 1982. Període orientat a destrucció.
- Entre 1983 i 1984. Període orientat a avaluació.
- Entre 1985 i l'actualitat. Període orientat a prevenció.

#### Fase 1. Orientació a debugging.

Durant aquest període, totes les proves estaven orientades a la correcció directa del codi font dels programes, essent realitzades directament pels propis desenvolupadors.

#### Fase 2. Orientació a demostració

Durant aquesta fase, el món de les proves principalment es basen en la realització de *checkouts* exhaustius, on es focalitzen en dos aspectes clau<sup>23</sup>:

- Debugging: assegurar que el programa funciona
- Testing: assegurar que el programa resol el problema per al qual s'havia realitzat.

En aquest moment, les proves es realitzaven al final del desenvolupament, de forma massiva per a garantir que es compleix amb l'especificació.

"La prova de software pot demostrar l'existència d'error, però mai no podria demostrar l'absència dels mateixos." E.W.Dijkstra.

<sup>22</sup> The Turing Digital Archive – "[Checking a large routine](#)"

<sup>23</sup> El 1957, C. Baker en su artículo "Review of D.D. McCracken's Digital Computer Programming" ja ho mencionava.

### Fase 3. Orientació a destrucció

En el llibre *The Art of Software Testing*, que G.J. Myers va editar l'any 1979, menciona que el "Testing és el procés d'executar un programa amb la intenció de trobar-hi errors". I aquest fet és un autèntic canvi en l'enfocament del testing ja que suposa canviar d'*intentar demostrar que un programa és correcte mitjançant proves i demostracions teòriques basades en matemàtiques, a intentar fer fallar el programa*.

És en aquest moment que els *tests* es converteixen en Casos de Prova (o Test Cases).

### Fase 4. Orientació a evolució

El 1983, la *Guideline for Lifecycle Validation, Verification and Testing of Computer Software*. NBS FIPS <sup>24</sup> proposa i descriu una metodologia que integra anàlisi, revisió i testing dins el cicle de vida del desenvolupament de software. I és en aquest moment que les proves de software comencen a integrar-se dins de les diferents metodologies de desenvolupament de software.

"L'objectiu general de les proves de software és confirmar la qualitat dels sistemes software executant sistemàticament el software en qüestió sota unes circumstàncies cuidadosament controlades". E.F. Miller

### Fase 5. Orientació a prevenció

En aquesta cinquena fase, que molts estudis consideren que va iniciar-se l'any 1985 quan H. Hetzel, i D. Gelperin, implementen STEP ( Systematic Test and Evaluation Process )sobre els estàndards formals IEEE 829 - 1983 y 828 - 1998. Aquest fet va aconseguir posar de major rellevància el testing de software dins el cicle de desenvolupament, obrint-se la possibilitat de integrar les proves en les diferents fases del mateix.

És en aquest moment que les proves es van diversificant per a cobrir totes les fases del desenvolupament, tots els artefactes, prototips, models, mòduls etc que componen el software del moment.

A partir d'aquest moment les proves s'aniran considerant una part clau de tot el cicle de desenvolupament; i el testing de software s'anirà desenvolupant de la mà de les novetats del propi desenvolupament de software, com per exemple, a partir d'introduir la Programació Extrema a partir del 1999, es començarà a perfilar el *Test Driven Development*, una metodologia de desenvolupament basada en proves.

---

<sup>24</sup> Abstract and Bibliography from NBS Special Publication 500-106, "[Guidance on Software Maintenance](#)"

### 8.1.3 Tipus de proves

Existeixen múltiples criteris i maneres diverses de classificar les diferents proves, segons l'àmbit concret en el que es realitzi aquesta classificació:

#### **Validació vs. Verificació:**

La Validació és un procés subjectiu on es valora si es correcte el que fa un sistema, és a dir, es centra en el resultat.

Per altra banda, la Verificació és un procés objectiu que es centra en el que se suposa que ha de fer un programa, comprovant que tot s'executa de forma correcta i que no s'executen funcions que poden degradar el funcionament.

#### **Formals vs. No Formals.**

Les proves Formal necessites d'una especificació formal del problema, i per tant, solen ser proves molt potents. Aquesta especificació és costosa i complexa, y sol requerir personal altament qualificat. Aquest tipus de proves solen aplicar-se en àmbits molt concrets, és a dir, zones crítiques de sistemes.

Les proves No Formals es basen en especificacions realitzades en llenguatge natura (especificacions funcionals, històries d'usuari, ...); són menys potents que les anteriors, però el seu ús està molt més generalitzat.

#### **Estàtiques vs. Dinàmiques.**

Les proves Estàtiques no requereixen de l'execució del programa; normalment es basen en la revisió del codi font i en la creació de traces d'execució. Solen ser manuals.

Les proves Dinàmiques requereixen l'execució del programa, execució directa o indirecta, tant si es realitza de forma manual com automatitzada.

#### **Caixa Blanca vs. Caixa Negra**

Les proves de Caixa Blanca comproven com un sistema s'executa, és a dir, es centren en el mecanisme intern del programa, comprovant pas a pas cadascuna de les accions del mateix.

Les proves de Caixa Negra, per contraposició, ignoren l'interior del programa, centrant-se únicament en les entrades i sortides del sistema (d'aquí el seu nom),

### Manuals vs. Automàtiques.

Les proves Manuals són les proves executades per un usuari. La fiabilitat sempre pot estar en dubte, ja que el factor humà en l'execució pot causar que es cometin errors

Les proves Automàtiques són aquelles proves que s'implementen i s'executen de forma automàtica. Són més fiable, però per altra banda, no sempre són possibles i en alguns casos el cost pot ser massa elevat.

#### 8.1.4 Nivells de proves

Tal i com s'ha mencionat, actualment el testing de software s'ha incorporat en totes les fases del cicle de vida del software, però lògicament, el testing que es du a terme en cada nivell de desenvolupament de software té un enfocament diferent i amb diferents objectius:

- **Proves Unitàries (Unit Testing)** : aquest tipus de testing té lloc al nivell més baix de proves. Està enfoca a provar les unitats bàsiques del software, és a dir, les part més petites de software que es poden provar. Aquests poden ser “unitat”, “mòduls” o “components”.
- **Proves d'Integració (Integration Testing)**: aquest nivell de proves té lloc quan dues o més unitats proves es combinen en una estructura més gran. És duen a terme per a comprovar el funcionament entre mòduls. Hi ha diverses estratègies per a executar-les, una incremental i l'altra no incremental.
- **Proves de Sistema (System Testing)**: aquest tipus de proves tenen com a objectiu avaluar el sistema en el seu conjunt, comprovant la qualitat del sistema sencer de principi a fi (end-to-end testing). El seu objectiu per tant, és el de comprovar que es compleixen els requisits funcionals i les especificacions tècniques del sistema. A més a més, en molts casos, ja que validen requisits no funcionals.

En aquest punt, habitualment les proves s'executen en un entorn el màxim de similar a com serà posteriorment a producció

- **Proves d'Acceptació (Acceptance Testing)**: aquest tipus de prova comencen a intervenir quan el sistema està (pràcticament) acabat i validat en fases prèvies; i en aquest moment és quan les proves del sistema les realitzen moltes vegades el propi client o els usuaris finals. El propòsit d'aquesta fase és que el client guanyi en confiança en el sistema que no trobar errors pròpiament.

## 8.2 Metodologia: Test Management Approach (TMap)

Dins el món del Testing, una classificació molt inicial i bàsica dels processos de proves es pot realitzar en base a si el testing que es realitza és **estructurat** o bé **no estructurat**.

El testing no estructurat l'entendem principalment com una situació de desordre, en la que no és possible predir l'esforç de proves, executar de forma factible les proves o mesurar el resultat de forma efectiva. En molts casos s'anomena "**ad hoc testing**". Aquesta aproximació té el problema que no empra cap criteri de qualitat, per exemple, per prioritzar correctament els riscos i les activitats de prova. Per altra banda, el testing estructurat aporta una sèrie d'avantatges, entre les que en podríem destacar les següents:

- Pot ser usat en qualsevol situació, independentment del client o de la metodologia emprada per a desenvolupar el producte.
- Aporta informació i recomanacions sobre qualsevol risc en referència a la qualitat del sistema sota prova
- Es tendeix a trobar els errors en una fase inicial
- Evita nous defectes
- Els productes resultats de les proves (ex. Casos de prova) són reutilitzables
- Tot el procés en el seu conjunt és comprensible i manejable.

En el nostre projecte des de bon principi vàrem optar per una aposta clara i concisa per aconseguir una alta qualitat del sistema a desenvolupar, fet que ràpidament va situar-nos dins el món del testing estructurat. Com s'ha comentat a sobre, el testing no estructurat té alguns avantatges interessants, i de fet, en certs moments s'ha dut a terme un testing més semblant a aquest per avaluar d'una primera passada l'estat d'alguns elements.

Actualment al mercat podem trobar moltes metodologies de testing estructurades, però des de bon principi l'opció per a que ens vam decantar és la de seleccionar **TMap (Test Management Approach)** com a metodologia de proves per al nostre projecte. Com és veurà en el següent punt és una de les metodologies més utilitzades al mercat (per exemple, és la metodologia més usada a Holanda i Alemanya), i gràcies al coneixement tant teòric com d'ús laboral en diversos projectes del que disposem, va semblar la millor opció.

### 8.2.1 Què és TMap?

TMap és la metodologia de proves desenvolupada per l'empresa Sogeti. TMap, o Test Management Approach, és una metodologia estructurada de proves per a sistemes d'informació.

La primera versió de la metodologia va ser definida l'any 1995, descrita per Martin Pol, Ruud Teunissen i Erik van Veenendaal. Cap a finals de 2006 una nova versió va ser publicada amb el nom de **TMap NEXT**, escrita per altres autors (Tim Koomen, Michiel Vroon, Leo van der Aalst i Bart Broekman). TMap va ser creada per la divisió holandesa de Sogeti, que forma part del grup Capgemini, i tot i que originàriament va ser un producte holandès, compta amb una extensa expansió al voltant del món, i s'ha traduït al francès, l'alemany i l'anglès.

TMap es pot resumir en quatre punts essencials que ho defineixen:



1. *TMap està basat en business driven test management, o una gestió del test enfocada al negoci.*
2. *TMap descriu un aproximació estructurada a les proves.*
3. *TMap conté una caixa d'eines completa.*
4. *TMap és un mètode de test adaptatiu.*

#### **Business Driven Test Management**

Aquest primer punt essencial està directament relacionat amb el fet que la importància de la relació entre IT i negoci en les empreses està en continu creixement. De cara al testing això representa que totes les eleccions sobre quins riscos caldrà cobrir amb el testing, quins resultats s'han d'entregar i quan temps i diners s'han de gastar estaran basades en uns valors i un coneixement racional i econòmic.

Aquesta és la raó per la qual TMap ha desenvolupat la gestió de proves enfocada al negoci, que pot ser vist com el "fil conductor" del procés de proves estructurat de .

### *Procés estructurat de proves*

En descriure un procés de prova estructurat (punt 2) i donant una completa caixa d'eines completa, TMap respon a les preguntes clàssiques què / quan, com, amb què i qui. Amb la descripció del procés d'ús de la prova s'ha realitzat el model del cicle de vida de TMap: un cicle de desenvolupament relacionat amb la descripció dels cicles de. El model del cicle de vida descriu què / quan s'ha de fer.

### *Caixa d'eines completa*

A més dels punts anteriors, per a ser capaç d'executar el procés de prova correctament, diverses punts en aspectes la infraestructura (amb el que), tècniques (com) i l'organització (qui) s'han d'implementar. TMap ofereix una gran quantitat d'informació aplicable en forma d'exemples, llistes de control, descripcions de la tècnica, procediments, estructures organitzatives per als tests i entorns de proves / eines (punt 3).

### *Mètode adaptatiu*

A més a més, TMap té un disseny flexible i que per tant, li permet ser aplicat a diverses situacions i diversos sistemes de desenvolupament. Per exemple, nous desenvolupament, manteniments de sistemes d'informació, desenvolupament *in-house* o també és aplicable a la compra de paquets externs, i s'adapta fins i tot a l'externalització de (parts) del testing.



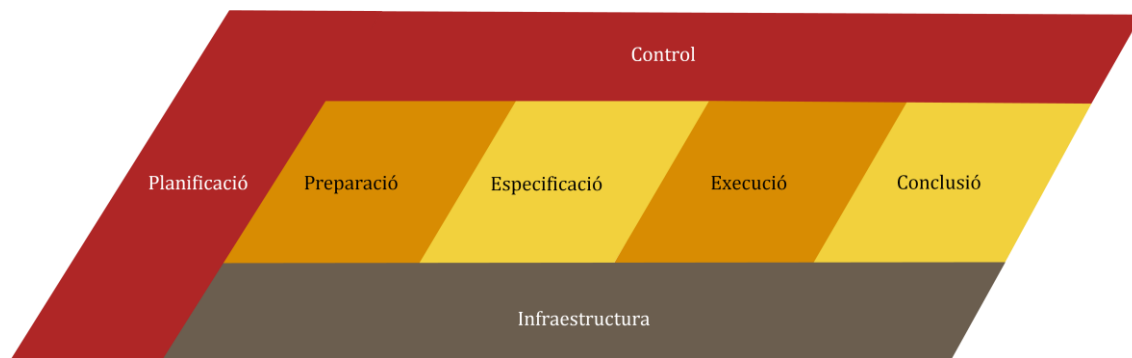
### 8.2.2 Cicle de vida segons TMap

Com qualsevol altre procés o metodologia de desenvolupament, un procés de proves consta d'un nombre de diferents activitats. Un model de cicle de vida de test és necessari per a estructurar les diverses activitats i el seu ordre, així com les seves dependències.

El model del cicle de vida és un model genèric; es pot aplicar a tots els nivells de prova i tipus de proves, i s'utilitza en paral·lel amb el cicle de vida propi del desenvolupament.

En el cicle de vida de TMap, les activitats de prova es divideixen en set grans fases, i cadascuna de les fases es divideix en una sèrie d'activitats:

- Planificació
- Control
- Infraestructura
- Preparació
- Especificació
- Execució
- Conclusió



Cal tenir en compte que la presentació de les diferents fases en el gràfic, per mitjà d'una paral·lelogram tort no és purament estètic, sinó que ve a representar que en realitat no hi ha cap fase que tingui lloc totalment disjunta de les que l'envolten, sinó que en molts casos les activitats tenen lloc en paral·lel (per exemple, en molts casos l'execució de les proves pot començar en alguna àrea abans d'acabar completament la fase d'especificació).

### 8.2.2.1 Planificació

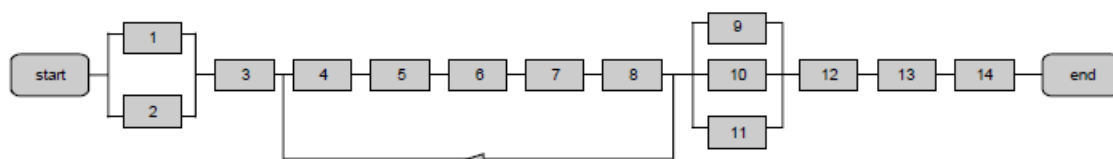
En primer lloc, i com en la majoria de metodologies, hi trobem la fase de **Planificació**. El cicle de vida de testing comença en el moment que el pla de proves i l'estratègia de proves s'escriuen, incloent-hi tota la informació necessària sobre el projecte en curs, un anàlisi de riscos, la planificació de les tècniques de test que s'usaran, una planificació a nivell de recursos de test, i una estimació.

Segons TMap, una fase de planificació per a un nivell de proves pot començar quan es disposa del següent:

- El sistema o client per al nivell de prova per al que es planifica
- L'objectiu final del producte
- Requisits generals del sistema
- Un equip i organització per als processos de desenvolupament
- El pla d'entregues del projecte
- El mètode que es farà servir per a desenvolupar el producte.
- Si existeix un pla de proves general (Master Test Plan), ha d'estar acabat i aprovat
- Informació sobre els entorns de desenvolupament i producció, per a poder definir l'entorn de proves.

Si es disposen de tots els elements mencionats, és pot començar amb la fase de planificació; i llavors, durant la creació del pla de proves, es duran a terme les següents activitats:

1. Establir les tasques (*Establishing the assignment*)
2. Entendre les tasques (*Understanding the assignment*)
3. Determinar la base de proves (*Determining the test basis*)
4. Anàlisi dels riscos del producte (*Analysing the product risks*)
5. Determinar l'estratègia de prova (*Determining the test strategy*)
6. Estimació de l'esforç (*Estimating the effort*)
7. Determinació de la planificació (*Determining the planning*)
8. Assignació unitats de test i tècniques de test (*Allocating test units and test techniques*)
9. Definir els productes de prova (*Defining the test products*)
10. Definir l'organització (*Defining the organisation*)
11. Definir la infraestructura (*Defining the infrastructure*)
12. Organitzar la gestió (*Organising the management*)
13. Determinar els riscos i contramesures del projecte de prova (*Determining the test project risks and countermeasures*)
14. Avaluació i consolidació del pla. (*Feedback and consolidation of the plan.*)

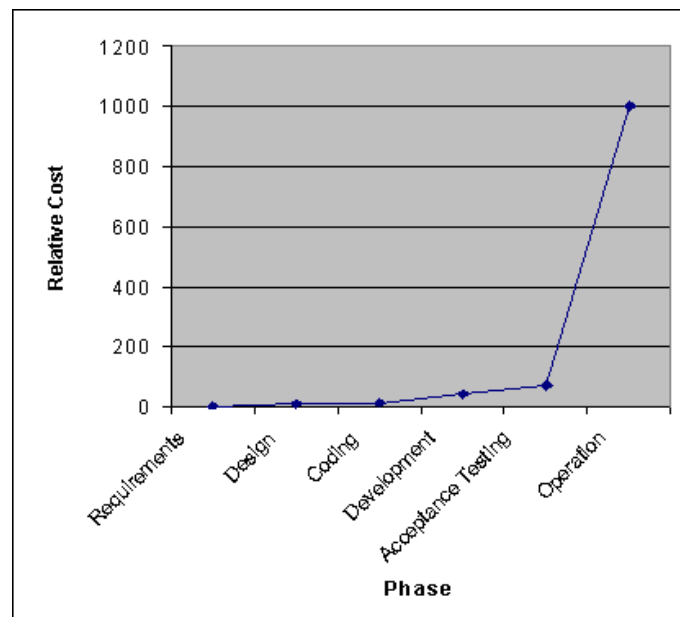


### 8.2.2.2 Preparació

Durant la fase de Preparació té lloc la revisió de tota la documentació que servirà de base per a les proves que es duren a terme. L'objectiu principal d'aquesta fase és tenir accés a una documentació (en anglès, s'usa el terme test basis en aquest punt) d'una qualitat adequada per a procedir, en la següent fase, a dissenyar els tests.

A més a més, una revisió primerenca de la base de proves millora la qualitat d'aquesta base i de la documentació, en detectar errors en un estat molt incipient, i per tant, evitar futurs errors que serien molt més costosos de corregir. Degut a que els equips de desenvolupament treballen en el desenvolupament del nou sistema en base a la documentació del sistema (que és part de la base de proves). Aquesta documentació pot contenir errors que poden causar una gran quantitat de – sovint costoses – feines de correcció posteriors si no es detecten de forma oportuna.

Un error detectat en aquesta fase és molt més barat i fàcil de corregir que posteriorment:



Font: <sup>25</sup>

<sup>25</sup> US Department of Transportation - [Developing Functional Requirements](#)

### 8.2.2.3 Especificació

Com el seu nom indica, la fase d'especificació *específica* les proves requerides i els punts de partida inicials per a l'execució de les proves. L'objectiu és preparar tant com sigui possible les proves, de manera que aquestes es puguin executar el més ràpidament possible a partir del moment en el que els desenvolupadors lliure l'objecte de prova.

Aquesta fase comença una vegada s'ha validat correctament les bases de prova (documentació, requisits, etc). I no és una fase que comenci un cop s'entrega el sistema o software per part dels desenvolupadors, sinó que és una activitat que té lloc en paral·lel, i a l'*ombra*, de la codificació del software.

Hi ha la falsa creença que únicament la part d'execució és la pròpiament important dins les fases de proves, ja que és la fase més visible. Però de mitjana únicament cobreix un 40% de les activitats de prova

Una interessant analogia d'aquest fet, és com mostra la imatge de la dreta, la semblança amb un iceberg. En aquest, sols una petita part és visible a la superfície, quan en realitat gran part es troba sota l'aigua. Doncs amb el testing passa una cosa similar: la part visible és l'execució, que s'emporta un 40% de les activitats de test; les altres activitats – planificació i preparació – poden representar un 20% i un 40% de l'esforç de testing, respectivament.

Hi ha moltes vegades que no es reconeix la importància que tenen aquestes dues parts; quan a més a més, si es poden dur a terme aquestes dues activitats tant abans com sigui possible de l'execució final de les proves, la contribució al temps total (al temps crític) del projecte per part del testing es veu reduïda.



## Conceptes bàsics associats al disseny de proves

### Situació de prova

Una condició aïllada en què l'objecte de prova mostra un comportament específic que necessita ser provat.

### Cas de prova lògic

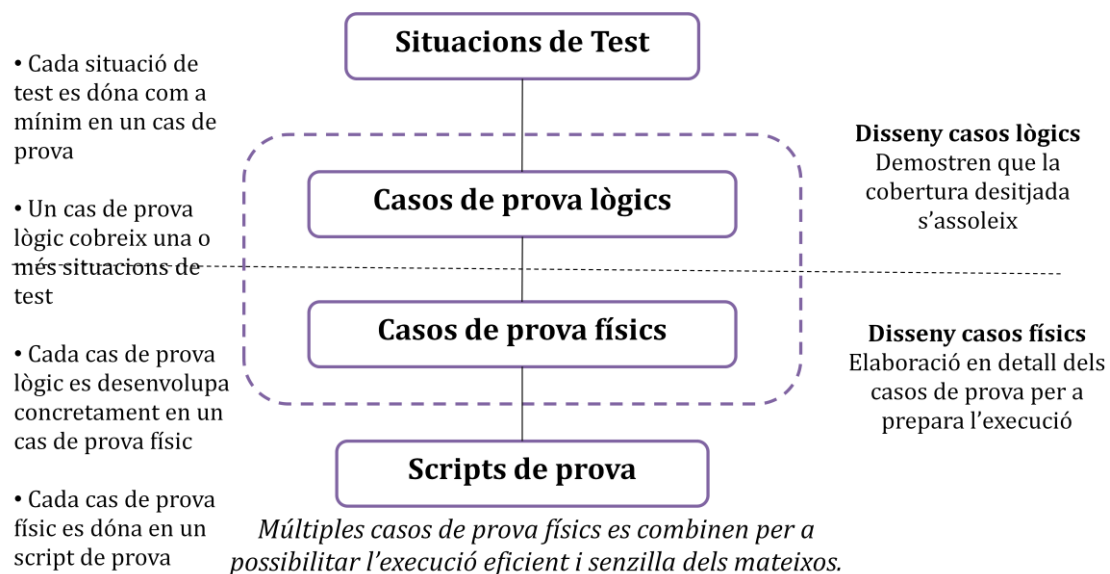
Describeix en termes lògics, les circumstàncies en què el comportament del sistema s'examina mitjançant la indicació que les situacions de prova estan cobertes pel cas de prova.

### Cas de prova físic

Elaboració concreta d'un cas de prova lògic, amb la selecció feta dels diferents possibles valors requerits per a l'entrada de dades i els paràmetres dels factors ambientals.

### Script de prova

Múltiples casos de prova físics es combinen per a ser capaços d'executar-los d'una manera eficient i simple

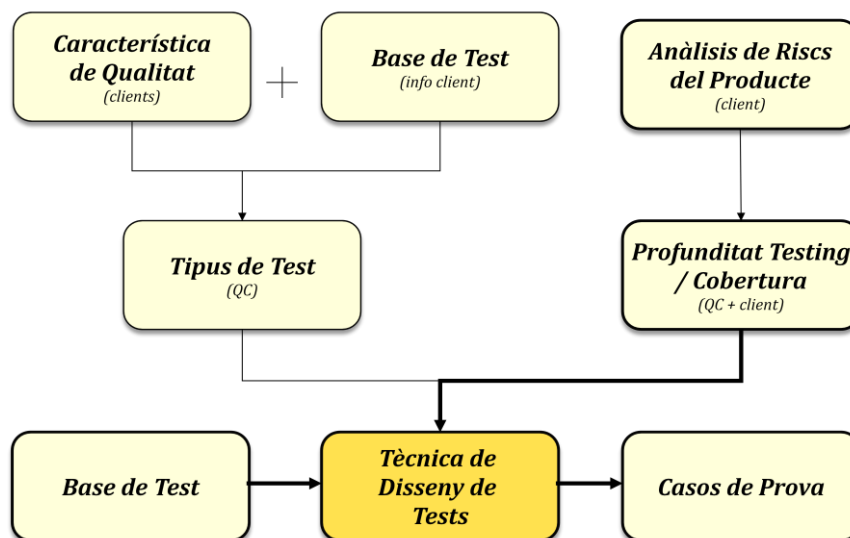


### *Tècniques de disseny de proves*

Les tècniques de disseny de proves són aquelles tècniques que s'utilitzen per a derivar els casos de prova necessaris per a cobrir la cobertura de proves requerida per a unes bases de test específiques.

Les tècniques de disseny de proves sorgeixen del fet que **no és possible provar-ho tot** dins uns paràmetres de temps i costos raonables especificats per als projectes.

La idea principal, i un dels nuclis de la metodologia TMap, és el fet que com més important és una funcionalitat, sistema, etc, més test intensiu requereix; i aquest fet està mapejat en un anàlisi de riscos del producte.



La importància i els principals beneficis d'utilitzar tècniques de disseny de test (TDT, de l'anglès, Test Design Techniques) són els següents:

- Les tècniques de disseny de tests proporcionen una elaborada justificació de l'estratègia de proves → a cobertura acordada en el lloc convingut.
- L'enfocament de les TDT a l'assoliment d'una cobertura específica per detectar un tipus específic de defectes → defectes que es detecten amb major eficàcia (en lloc de les proves ad hoc)
- Les proves són reproduïbles, el procés de prova és independent de la persona que especifica i executa, i les especificacions de prova són transferibles i mantenibles.
- Amb l'ús de les tècniques de disseny de tests, la planificació i gestió del procés de testing és més fàcil → l'especificació de les proves i l'execució de les mateixes són blocs separats.

#### 8.2.2.4 Execució

El principal objectiu de la fase d'execució és conèixer millor i obtenir informació sobre la qualitat de l'objecte o sistema que es troba sota prova mitjançant l'execució de les proves acordades.

L'execució real de la prova s'inicia quan l'objecte de prova, o una part separada del sistema que sigui provable, s'entrega. En primer lloc, el software es revisa per a comparar-ne la completesa (d'aquesta forma, si l'entrega ja no és correcta, l'equip de test no perd el temps en validar una versió incorrecta). Si la revisió es superada satisfactòriament, llavors s'instal·la en l'entorn de proves seleccionat. I un cop aquí encara no es comença directament amb les proves, sinó que s'executa un "pre-test" de que valida de forma bàsica si el sistema d'informació que s'ha de prova, en combinació amb la infraestructura de proves, tenen suficient qualitat inicial per a proves exhaustives.

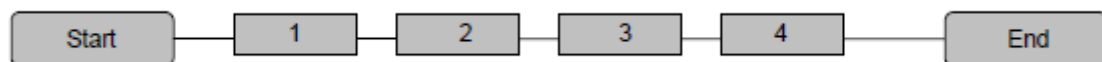
Una vegada s'ha donat el vist-i-plau bàsic a l'entorn amb la nova versió, es pot procedir a l'execució de les proves prèviament dissenyades en l'anterior fase. Durant l'execució és importar emmagatzemar tots els resultats que es van obtenint, per a posteriors mètrics i per a tenir un coneixement més exacte de l'estat de l'aplicació, així com de l'avanç de les proves.

Durant l'execució principalment el que es fa és comparar les diferències, si n'hi ha, entre els resultats esperats i els resultats que s'obtenen. I en casos que difereixin ambdós resultats, acaba provocant el registre d'un nou defecte o incidència.

Per tant, tal i com s'ha explicat, durant la fase d'Execució tenim les següents activitats:

1. Validació i acceptació de l'objecte de proves
2. Preparació del punt inicial del sistema (si es requereix d'algunes modificacions, ja sigui a nivell de dades del sistema, ...)
3. Execució (o reexecució) de les proves
4. Revisió i avaluació dels resultats de les proves.

El següent esquema mostra la seqüència i les dependències entre les diverses activitats:



### Gestió d'errors

Un altre aspecte rellevant de mencionar dins de la fase d'Execució, tot i que no és exclusiu d'aquesta fase (però sí que és en la que adquireix més rellevància) és el de la **gestió d'errors**.

Tot i que errònia, existeix en molts casos la creença que el trobar errors és el principal objectiu de la fase de proves. Ha de quedar clar que el propòsit del testing va molt més enllà que aquest punt: donar la informació i recomanació sobre els riscos i la qualitat, ...

Però per altra banda, el fet de trobar errors no deixa de ser una de les activitats més importants del testing.

En molts casos es poden utilitzar diverses paraules per a referir-se al mateix concepte, o similars, com poden ser error, defecte, incidència, fallada, etc. Però segons TMap, la definició a utilitzar en referència a un defecte és que **és el resultat d'un error resident en el codi o la documentació**.

Com s'ha comentat és durant la fase d'execució que principalment poden detectar-se errors, però en la Preparació i Especificació també se'n poden trobar, tot i que en aquests casos els errors no seran pròpiament a l'objecte de proves (ja que encara no està ni disponible), sinó que seran a la documentació i a la base de proves.

Quan es troba qualsevol defecte, els passos que ha de seguir el responsable de les proves són els següents:

1. Recopilació d'evidències
2. Reproducció de l'error
3. Verificació d'errors propis en la prova
4. Determina la pressuposada causa externa de l'error
5. Isolar la causa (opcional)
6. Generalitzar el defecte
7. Comparació amb altres defectes
8. Escriure un informe d'error

#### Informe d'errors:

Un informe o report d'error és més que una simple descripció del defecte. Altres detalls a part de la descripció poden ser requerits; i per a fer-ho d'una forma estructurada, l'informe s'acostuma a dividir en diversos *camp*s.

El mínim nombre de camps que es recomana en un informe de defectes són els següents:

1. **Projecte o nom del sistema:** com indica, el nom del projecte o del sistema de prova
2. **Identificació única:** un identificador únic, normalment un seqüencial alfa numèric, que serveix per a gestió i identificació.



3. **Breu descripció:** Una breu caracterització del defecte en un nombre limitat de paraules, màxim una frase, que preferiblement també indica clarament les conseqüències del defecte.
4. **Qui ha introduït el defecte:** nom de la persona que hagi obert el defecte.
5. **Identificació del nivell de fase / test :** la fase o nivell de prova en què es va trobar el defecte, per exemple, disseny, proves de sistema, ...
6. **Severitat:** La categoria de gravetat proposada pel tester. Aquesta categorització ha de reflectir el dany que suposa a nivell de negoci l'error. En cada empresa i projecte es poden tenir diverses categoritzacions, però a nivell general sempre es poden separar en:
  - a. **Crítica:** implica alts costos i problemes en operacions de producció (pèrdues de vides humanes, negoci, ...)
  - b. **Greu:** els costos involucrats no són tant alts, i poden implicar que l'usuari ha de refer part de la feina
  - c. **Perjudicial:** pocs costos involucrats, o qüestions relacionades amb la facilitat d'ús
  - d. **Cosmètica o Estètica:** mal disseny que no és un problema per al client extern, però pot ser molest internament.
7. **Prioritat :** La prioritat de la solució proposada pel tester. Una possible classificació general pot ser la següent:
  - a. **Correcció immediata** (per exemple, es requereix d'una correcció disponible en les pròximes 48 hores que resolgui el problema): en aquest cas o el procés de proves o bé els processos productius es veuen seriosament obstruïts.
  - b. **Correcció dins la versió actual:** el procés actual pot continuar funcionant amb solucions temporals, si cal, però a nivell de producció no s'hauria de sobrecarregar amb aquests problemes.
  - c. **Correcció en la propera versió:** el problema (en el moment de la detecció de l'error no apareix a producció, o en cas contrari el dany causat és lleu
8. **Causa:** el tester indica on creu que es troba l'error.
9. **Identificació de l'Objecte de Proves:** s'indica en quin objecte de proves s'ha localitzat l'error, i a ser possible, en quina part del mateix
10. **Cas de prova:** referència al cas de prova que s'ha executat, i durant l'execució del qual s'ha detectat l'error en qüestió.
11. **Descripció del defecte:** descripció del defecte de forma tant precisa com sigui possible
12. **Estat:** el diversos estats del cicle de vida del defecte que es contemplen en el projecte. Per exemple, pot ser "Nou", "En procés", "Rebutjat", "Solucionat", "Fet", ...

En alguns casos aquesta llista es pot ampliar amb altres camps possibles, com la identificació de l'entorn de proves, la identificació de la base de test, etc. Però com es pot observa, tot i la recomanació de que tot informe d'errors ha de contenir com a mínim els camps obligatoris, el nombre camps és força gran i en molts casos, es tendeix a una simplificació dels informes d'errors.

### 8.2.2.5 Conclusió

L'enfocament estructurat per a les proves de TMap pot reportar molts beneficis per a la **repetibilitat del procés**. Permet que els productes puguin ser reutilitzats en les proves posteriors, si compleixen amb certs requisits. Aquest fet pot permetre accelerar certes activitats.

Els productes poden ser coses tangibles, com els casos de prova o entorns de prova (**testware**), però també coses no tangibles com l'experiència (**avaluació del procés**). En preservar el *corpus* de proves, es realitza una selecció ja que sovint aquest corpus pot contenir grans quantitats d'elements interessants a guardar. L'avantatge de preservar testware és perquè permet actualitzar amb un esforç reduït quan es canvia el sistema per executar una prova (regressió), per exemple. No hi ha per tant una necessitat de (re)dissenyar una nova prova.

El procés de la prova també s'avalua en aquesta fase. L'objectiu és aprendre de l'experiència adquirida i aplicar aquestes lliçons apreses en una nova fase de proves, si n'hi ha. Aquest procés també serveix com a entrada per a l'informe final, que el mànager de qualitat crea durant la fase de control.

### 8.2.2.6 Control

Com en molts del projectes, no únicament de test sinó de desenvolupament o fins i tot en general, el projecte s'executa rarament exactament segons el previst. I per tant, la execució del pla de proves també ha de ser controlada i ajustada, si s'escau. I aquest punt és l'objectiu de la **fase de control**. Tal i com havíem vist en la introducció a la metodologia de TMap, la fase de control juntament amb la de infraestructura són les dues úniques fases que s'estenen a partir de la Planificació i fins que acaba la Conclusió; i això és bàsicament perquè el que ens interessa controlar és tot el procés de proves, i no únicament una de les subfases com podria ser l'execució.

L'objectiu de les activitats d'aquesta fase és **controlar i informar sobre el procés de proves** d'una manera òptima, de manera que el client té una visió adequada i el control sobre el progrés i la qualitat de el procés de prova i la qualitat del sistema que s'està provant.

El mànager de les proves gestiona el procés complet de proves, la infraestructura i els productes que es volen provar. Basant-se amb les dades que pot obtenir durant la fase de Control, el mànager pot analitzar possibles tendències. S'ha d'assegurar d'estar correctament informat dels desenvolupaments que estan en curs, per a conèixer endarreriments, grans canvis previstos, i ajustaments generals en la planificació de tot el projecte.

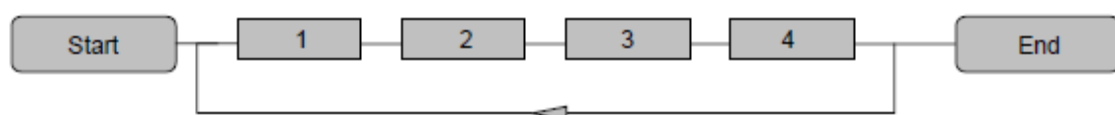
I donat el cas, si es veu necessari, el mànager de les proves pot proposar mesures de mitigació i correcció al client.

**La informació és el producte principal del testing.** Amb aquesta finalitat, el mànager de la fase de proves pot crear diferents tipus d'informes per als diferents grups de destinataris, tenint en compte dels aspectes que el "Business Driven Test Management" ens aporta en termes de resultat, riscos, temps i cost.

Com s'ha mencionat abans, aquesta fase de "Control" no comença just al principi, però es considera que comença just després de la creació del Pla de Proves, i les seves principals activitats són:

- 1 Gestió: gestió de tot el procés de proves, la infraestructura i el *test ware*
- 2 Monitorització: seguiment, recollició de informació sobre la qualitat i el progrés i identificació de riscos.
- 3 Informes (*Reporting*): creació dels informes requerits
- 4 Ajustaments: optimització del procés de proves

Que venen a executar-se de forma seqüencial al llarg del projecte (tot i que realment, en alguns casos els diferents punts s'executen de forma paral·lela):



### 8.2.2.7 Infraestructura

Si en l'apartat de Control comentàvem que la fase començava just en acabar el Pla de Proves, i s'estenia fins a la finalització del projecte, en aquesta fase ens trobem en un cas similar, ja que la fase de **Infraestructura** té com a tret de sortida quan els requeriments de infraestructura s'han establert i especificat en el Pla de Proves o bé en el Màster Test Plan.

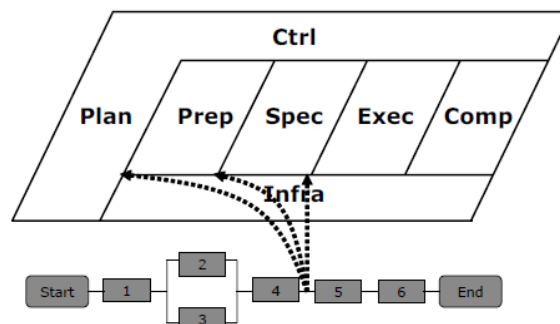
El principal objectiu d'aquesta fase és el fet de proveir al infraestructura de proves necessària, que serà utilitzada en diverses fases i activitats segons TMap. El concepte de infraestructura de proves consisteix en les instal·lacions i recursos necessaris per a facilitar l'execució satisfactòria de les proves; i en aquest cas, cal fer una distinció clara entre tres dels elements principals, que en molts casos poden requerir d'esforços (econòmics i temporal) diferents:

- **Entorns de prova:** tota la infraestructura que facilita pròpiament l'execució de les proves (alguns exemples: servidors, xarxes, bases de dades, etc).
- **Eines de prova:** eines que donen suport a l'execució de les proves. Com s'ha mencionat en l'apartat 3, per al projecte s'ha seleccionat una sèrie d'eines de gestió de proves, automatització, etc, que permeten realitzar les activitats de proves més àgilment.
- **Llocs de treball:** finalment, és considera també part de la infraestructura els llocs de treball i elements necessaris per a cada treballador.

La base de la fase de creació de la infraestructura i el manteniment es defineix en la fase de planificació. Aquí, dins de l'activitat "Definició de la infraestructura", és descriu de forma global la infraestructura necessària, incloent-hi la necessària per a la planificació. Aquesta descripció (del pla de proves mestre o del pla de proves), serveix d'entrada per a la primera activitat d'aquest fase.

La fase de creació i manteniment de la infraestructura consisteix en les següent activitats:

1. Especificació de la infraestructura
2. Clara comprensió de la infraestructura necessària
3. Especificació de la prova inicial de la infraestructura
4. Execució de la prova inicial de la infraestructura, per a validar-la
5. Manteniment de la infraestructura
6. Preservació de la infraestructura



### 8.3 Aplicació de TMap en el Projecte

TMap, tal i com s'ha explicat ens ofereix el marc metodològic de treball necessari per afrontar la tipologia de projectes que sigui necessari.

En el projecte actual, s'ha optat per a realitzar una breu simplificació ja que per tal i com s'ha treballat conjuntament amb el company, i en ser únicament dues persones en l'equip no ens era necessari plantejar-nos realitzar tots els passos.

En els propers apartats s'anirà presentant, ja de forma detallada i específica al projecte en si, com s'han anat realitzant les diferents fases, explicant en tot moment si s'ha usat algun tipus de documentació extra, eines, etc.

Principalment, les proves realitzades durant aquest projecte han estat Proves de Sistema (System Tests), que són les que s'aniran presentant a continuació.

### 8.4 Fase de planificació

Tal i com marca TMap, es va començar amb la fase pròpiament de planificació del Testing. Des del principi, l'abast de les proves a realitzar van quedar molt definides en la fase de proves de Sistema (System Test), proves a nivell funcional principalment, i enfocades a provar les diverses aplicacions que conformen SetelSys (Core, Maiden, Runner, ...) així com les eines d'ajuda (Chrono Checker, BD Comparer, ...).

#### 8.4.1 Pla de Proves o Test Plan

Des de la mateixa pàgina web de TMap es disposa d'una sèrie de plantilles per a la creació del Pla de Proves, i es va utilitzar aquesta plantilla per adaptar-la i crear el pla de proves concret per a SetelSys.<sup>26</sup>

En aquest document es va anar consolidant tota la informació rellevant a nivell de testing de sistema; fet que el va convertir en un document molt interessant en contenir una bona compilació de informació.

Sense la intenció de presentar tot el Pla de Proves, que per extensió no seria molt factible, si que interessa mostrar alguns dels apartats, per a veure'n el detall:

---

<sup>26</sup> TMap NEXT Downloads – [Template Test Plan System and Acceptance Tests](#)

Per exemple, es documenta quin és l'objectiu del pla de proves, i a més a més, tota la informació sobre el client, membres que formen part dels equips, ... :

### 1.1 Objectiu del pla de proves

L'objectiu d'aquest pla de proves (TP) per al nivell de Proves de Sistema és el d'informar a tots els que estan involucrats en el procés de la prova sobre l'enfocament, les activitats i les prestacions relatives a la prova del sistema per al projecte SetelSys. Aquest pla descriu l'elaboració concreta i detallada dels punts que habitualment s'inclourien en el Master Test Plan, però que en el cas d'aquest projecte, directament s'inclouran en el Pla de Proves de la fase.

També es detalla clarament l'abast de les proves, i quins apartats queden fora de la responsabilitat de l'equip de proves:

#### 1.2.4 Abast

SetelSys és un sistema de gestió de tots els procediments i aplicacions que actualment s'utilitzen en l'empresa, funcionalitats que no ofereix el seu sistema principal de gestió o ERP.

Dins l'abast: dins dels punts inclosos sota la responsabilitat de l'equip de proves, hi trobem les següents:

- Proves de Sistema dels diferents components que componen SetelSys (SetelSys Web, SetelSys Core, SetelSys Maiden, SetelSys Runner, SetelSys Services).
- Proves d'integració dels diversos components, incloent-hi proves de principi a fi, o end-to-end.
- Validació del sistema en un entorn de pre-producció
- Validació de les interaccions amb els seus sistemes actuals

Fora de l'abast: queda fora de l'abast del projecte de proves actual els següents punts:

- Proves unitàries dels diferents desenvolupaments
- Proves de rendiment
- Proves de seguretat
- Proves sobre el seu sistema actual

I també es defineixen en detall qualsevol punt de control que s'hagi de tenir en compte al llarg de l'execució de les fases de prova. Per exemple, aquí podem veure els punts que s'han tingut en compte a l'hora de controlar els passos entre les fases d'Especificació i d'Execució:

#### 4.4 Criteris d'entrada i sortida

A continuació es descriuen els criteris d'entrada i sortida establerts per les fases d'Especificació i Execució:

##### **Criteri d'Entrada per a la fase d'Especificació:**

- La documentació generada a nivell de requeriments ha estat aprovada per part del client.
- S'ha realitzat una primera revisió per part de l'equip de proves de la documentació, i s'ha considerat suficient per a procedir a l'especificació de proves.
- L'eina de gestió de proves està operativa, amb el Pla de Proves creat.
- Els requisits s'han carregat a l'eina de proves

##### **Criteri d'entrada per a la fase d'Execució:**

- Els casos de prova s'han creat correctament, amb la suficient cobertura de proves.
- Els casos de prova s'han carregat a l'eina de gestió de proves
- L'entorn de test està correctament instal·lat i operatiu.
- S'ha instal·lat la última versió de software de SetelSys
- S'ha realitzat un verificació inicial de l'entorn i l'aplicació per veure que els punts basics funcionen correctament.

##### **Criteri de sortida de la fase d'Execució:**

- Els casos de prova s'han executat i els resultats s'han emmagatzemat correctament a l'eina de gestió.
- S'ha executat un mínim del 90% dels casos de prova, i un 100% dels casos de prioritats 1 i 2
- No queden errors registrats oberts que siguin de prioritat 1
- No queden errors registrats oberts que siguin de prioritat 2, que no tinguin una solució alternativa (*workaround*)
- Els errors de prioritats 3 i 4 que restin encara oberts, s'han documentat correctament i el client n'està al corrent.

### 8.4.2 Eina de gestió de proves – Fase de Planificació

En l'apartat 3, on es detallen quines són les eines que s'han usat durant la realització d'aquest projecte, en quan a eines de Gestió de Proves, és a dir, eines necessàries per a gestionar tot el que fa referència a les proves, ja s'apuntava al fet que s'utilitzaria el **Microsoft Test Manager** (MTM).

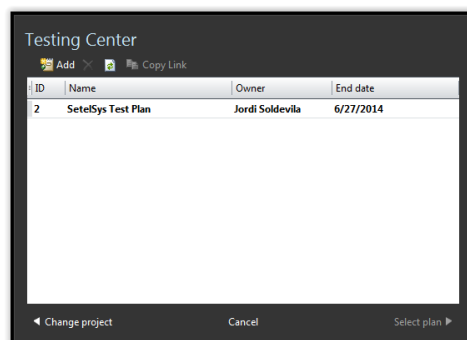
Pel simple fet d'utilitzar la suite de Visual Studio en el desenvolupament, l'opció més lògica era utilitzar directament la eina que Microsoft ja integra en algunes de les versions del Visual Studio, com és el Test Manager.

Aquesta eina s'adequava perfectament a les necessitats i als principals objectius requerits en el projecte:

- **Gestió completa de la fase de proves:** permetre'ns gestionar des del principi fins al final els nostres requeriments, proves i fases d'execució.
- **Traçabilitat:** tenir una clara traçabilitat de tots els elements del projecte (des dels requeriments fins a les proves executades, i els defectes si és el cas).
- **Integració:** una integració senzilla amb la resta d'eines seleccionades per al desenvolupament
- **Automatització:** possibilitat de vincular les proves automàtiques directament amb l'eina de gestió de les proves.

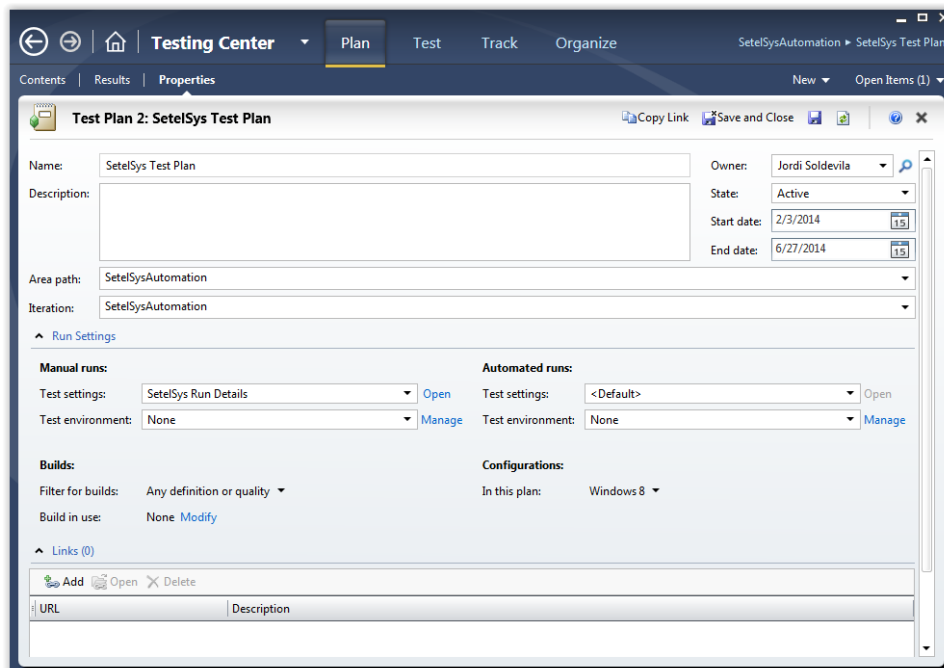
Durant la fase de Planificació, estrictament parlant encara no era imprescindible la utilització de l'eina de gestió de proves, ja que el pla de proves íntegre es va crear seguint la plantilla word de TMap. No obstant això, i aprofitant que aquesta part de la infraestructura d'eines ja la teníem en marxa, vam optar per a crear el "Test Plan" a l'eina, i configurar els detalls bàsics que en formarien part:

Inicialment, es va crear el "SetelSys Test Plan", el pla de proves a l'eina que contindria tots els elements posteriors:

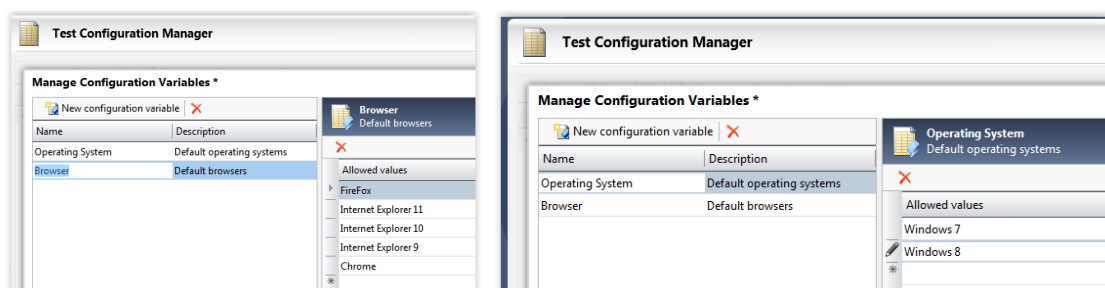




I posteriorment, es va procedir a detallar les propietats d'aquest pla de proves (dates del projecte, entorns de proves, etc ...)



I un detall a mencionar, en aquest moment també es van crear les diferents possibles combinacions que es podrien utilitzar a l'hora d'executar les proves i verificar el funcionament de l'aplicació. Per a fer-ho, es van tenir en compte les variables *Sistema Operatiu* i *Navegador*, per a crear diverses configuracions que a l'hora d'executar permetien seleccionar les més adequades:



En aquestes imatges es poden veure la creació de les diferents variables, que posteriorment es combinaven per a fer les diferents configuracions disponibles durant les proves:

Test Configuration Manager						
Manage configuration variables						
ID	Name	Default	State	Configuration variables	Description	
7	Windows 7 + FF	No	Active	Browser: FireFox Operating System: Windows 7		
10	Windows 7 + IE 10	No	Active	Browser: Internet Explorer 10 Operating System: Windows 7		
9	Windows 8 + Chrome	No	Active	Browser: Chrome Operating System: Windows 8		
6	Windows 8 + IE11	Yes	Active	Browser: Internet Explorer 11 Operating System: Windows 8	Default operating system for testing	

## 8.5 Fase de preparació

La fase de preparació, tal i com s'ha explicat en el seu apartat corresponent, té com a gran objectiu la recopilació i revisió de la base de proves.

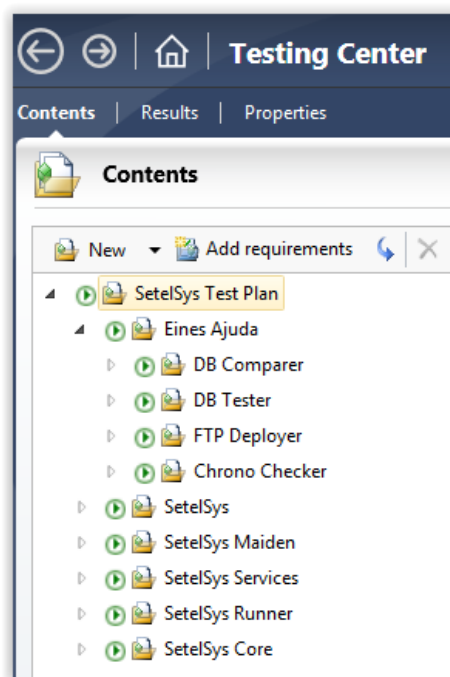
Durant la planificació general del projecte (apartat 4), a l'hora de fer l'Anàlisi de Requeriments ja es va plantejar el realitzar una sèrie de reunions amb el client per tal de clarificar i anar concretant amb més detall com hauria de ser el sistema. Tot i aquesta revisió, durant la fase de preparació es va procedir a una revisió en més detall, on es van detectar una sèrie d'errors menors i incoherències; fet que va permetre corregir-los en una fase molt incipient.

En el nostre cas, en aquesta fase li vàrem afegir més valor, ja que al realitzar la compilació de tota la documentació també es va procedir a la càrrega d'aquests requisits a l'eina de gestió de proves. El motiu principal per a fer-ho va ser pel fet de poder mantenir clarament una traçabilitat completa, des dels requisits, als casos de prova, a les diferents execucions i resultats, així com també als defectes corresponents a aquell requisit.

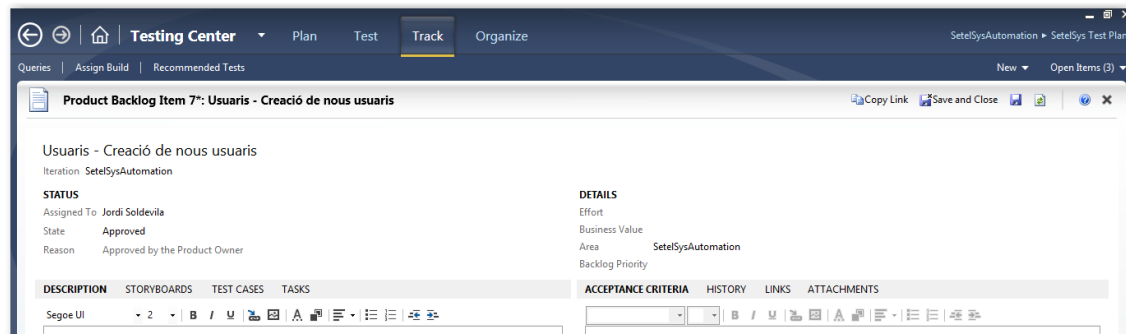
### 8.5.1 Eina de gestió de proves – Fase de Preparació

Tal i com es ve explicant, el Microsoft Test Manager (TMT) és la eina de gestió de proves escollida per a gestionar tot el procés de testing de l'aplicació.

Tal i com es pot veure en la imatge de sota, les proves s'han organitzat segons les diferents aplicacions de les que consta el sistema SetelSys, i cadascuna de les aplicacions contindrà, en la següent fase de testing, els casos de provar per a cobrir correctament les diferents àrees.

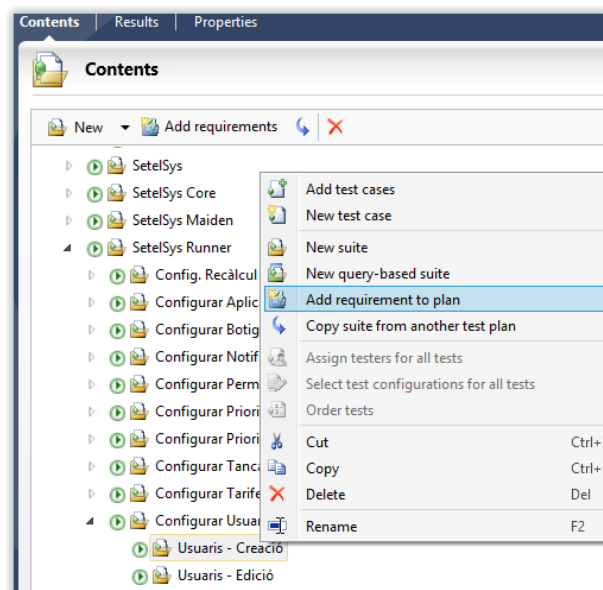


En quant a la creació dels diferents requisits a l'aplicació i la seva traçabilitat, en primer lloc s'ha procedit a anar creant els diferents requisits (a alt nivell, afegint el detall en la seva descripció). Per a fer-ho, s'ha usat la funcionalitat de nou "product backlog item", de la plantilla que s'ha usat del TFS i el Test Manager, enfocada a temes d'Agile, però que ja ens servien correctament:



Peu: exemple de creació de requisit de creació de nous usuaris

Una vegada creats els diferents elements, es va procedir a enllaçar cadascuna de les diferents test suites (o carpetes) amb els corresponents requisits:



Seleccionant els requeriments necessaris:

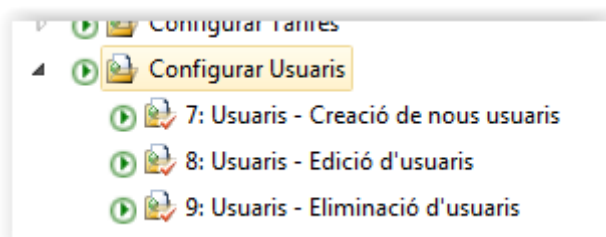
### Add existing requirements to this test plan

And/Or	Field	Operator	Value
►	Work Item Type	In Group	Requirement Category
And	Team Project	=	@Project
And	Area Path	Under	SetelSysAutomation

The following 3 items are available for selection.

ID	Title	State	Assigned To	Iteration Path	Area Path
7	Usuaris - Creació de nous usuaris	Approved	Jordi Soldevila	SetelSysAutomation	SetelSysAutomation
8	Usuaris - Edició d'usuaris	Approved	Jordi Soldevila	SetelSysAutomation	SetelSysAutomation
9	Usuaris - Eliminació d'usuaris	Approved	Jordi Soldevila	SetelSysAutomation	SetelSysAutomation

I finalment, disposant de les *test suites* corresponents a cada un dels requeriments



En fer aquest procediment, i tal i com veurem en l'especificació, a l'anar afegint els diferents casos de prova per a cobrir aquests requeriments, directament aniran quedant enllaçats amb els seus requisits, fet que permetrà controlar correctament la mencionada traçabilitat.

## 8.6 Fase d'especificació

La fase d'especificació del projecte SetelSys ha estat un dels punts més complexos a treballar, degut a la varietat de components que integra l'aplicació, fet que ha comportat la generació d'una gran quantitat de proves.

A mesura que es van anar tancant les especificacions i dissenys tècnics, i un cop realitzada la planificació de la fase de proves, així com al revisió en la fase de Preparació, es va procedir, funcionalitat per funcionalitat ha crear els casos de prova necessaris per a tenir un nivell cobertura de proves correcte per a la qualitat esperada.

S'ha optat per a no plasmar en al documentació una llarga llista de casos de prova dissenyats, sinó que en tots els exemples que es poden veure en el moment d'explicar com funciona l'eina de gestió de les proves, es podran veure algunes de les proves detalls.

Com a punt interessant a mencionar, és el fet que en tot moment s'ha intentat seguir la metodologia TMap en quant a l'especificació de proves, ja que dins de la metodologia podem trobar-hi una bona quantitat de tècniques de disseny de proves que simplifiquen la creació de les mateixes, garantint la cobertura esperada.

Algunes de les tècniques que s'han utilitzat per a derivar els casos de prova són les següents:

- **Anàlisi dels valors límits:** si el comportament del sistema canvia tan aviat com el valor d'un paràmetre excedeix d'un valor en particular, a aquest valor se l'anomena valor límit. I el fet de controlar aquest valors ens dona una important seguretat ja que és un dels punts més freqüents d'error.
- **Proves Sintàctiques:** tècnica per a comprovar la funcionalitat, així com la facilitat d'ús de l'aplicació, que requereix del coneixement de les entrades i sortides del sistema, així com de la descripció dels seus atributs.
- **Proves Semàntiques:** tècniques per a preparar les proves i validar la funcionalitat, tenint en compte també l'especificació de les entrades i sortides del sistema, però considerant regles de negoci que apliquin a la funcionalitat.,
- **Testing exploratori:** el testing exploratori es va fundar i es descriu com un concepte i un enfocament una sèrie d'anys enrere per part de James Bach. D'acord amb la seva definició el testing exploratori és l'aprenentatge simultani, el disseny i execució de les proves. És a dir cada tipus de prova en què el provador dissenya les seves proves durant la pròpia execució i la informació obtinguda es torna a utilitzar per dissenyar proves noves i millorades.
- Etc...

### 8.6.1 Eina de gestió de proves – Fase d'Especificació

És a partir de la fase d'especificació quan es posa en rellevància la necessitat de tenir una eina de gestió de les proves, ja que durant tot el projecte, s'han creat un nombre elevat de casos de prova que si no fos amb una eina d'aquest estil, seria impossible de gestionar.

Tal i com s'explica en l'apartat anterior de la Preparació, en aquell punt ja es van posar les bases de l'ordenació dels diferents jocs de proves que es van crear per a les proves de SetelSys. Tal i com es veia, es van separar segons les diferents aplicacions, vinculades als requisits per a tenir la traçabilitat.

Degut a la impossibilitat de presentar tots els casos de prova dissenyats per a les proves, s'ha optat per presentar la manera com es van dissenyar totes les proves, posant alguns exemples diversos, en els que es pugui veure tant l'ús de l'eina, com dels diferents paràmetres que es poden tenir en compte. I a la vegada, d'aquests casos aquí presentats, se'n mostra l'execució tant a nivell manual (en algun casos també, per a mostrar alguns dels errors detectats durant el procés de proves), i també es podrà veure la manera de validar els mateixos casos de prova de forma automàtica en l'apartat corresponent.

El Microsoft Test Manager ens permet, una vegada seleccionada la *suite* a la que volem afegir una nova prova, omplir un formulari amb tota la informació necessària per a poder crear la prova.

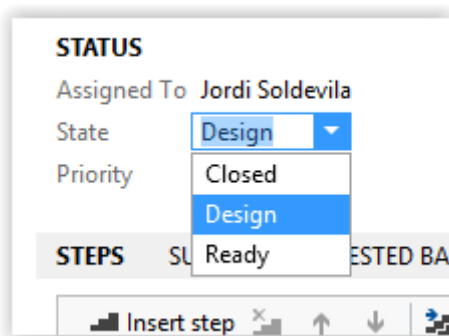
The screenshot displays the Microsoft Test Manager 'Testing Center' interface. The 'Plan' tab is selected, showing a test case titled 'Test Case 11: Nou usuari - Nom usuari - Obligatori'. The test case is associated with the 'SetelSysAutomation' iteration. The status is 'Assigned To: Jordi Soldevila', 'State: Design', and 'Priority: 3'. The details section shows 'Automation status: Not Automated' and 'Area: SetelSysAutomation'. The 'STEPS' tab is active, showing a list of 7 steps for creating a new user. The steps include actions like 'Executar SetelSys Runner', 'Introduir les dades correctes i clicar a [Entrar]', 'Clic a [Mainteniment]', 'Clic a [Configurar Usuaris]', 'Click a [Nou Usuari]', 'Omplir tots els camps per a crear un nou usuari menys el camp del "Nom Usuari"', and 'Clic a [Guardar]'. The expected results for each step are listed on the right. The interface also includes tabs for 'SUMMARY', 'TESTED BACKLOG ITEMS', 'LINKS', 'ATTACHMENTS', and 'ASSOCIATED AUTOMATION'.

STEPS	SUMMARY	TESTED BACKLOG ITEMS	LINKS	ATTACHMENTS	ASSOCIATED AUTOMATION
1.	Executar SetelSys Runner				
2.	Introduir les dades correctes i clicar a [Entrar]				
3.	Clic a [Mainteniment]				
4.	Clic a [Configurar Usuaris]				
5.	Click a [Nou Usuari]				
6.	Omplir tots els camps per a crear un nou usuari menys el camp del "Nom Usuari"				
7.	Clic a [Guardar]				

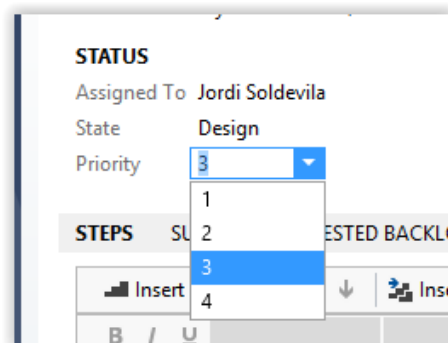
Ja es comentava en un punt anterior, que la gràcia de tenir unes proves ben definides permetia després una execució i una re-execució (si és dóna el cas) més senzilla i ràpida. A la vegada, el fet de complimentar al detall els casos de prova permet que, en el futur, la persona encarregada d'executar les proves sigui diferent al que les ha especificat, i que pugui fer-ho sense que el cost de traspàs de coneixements sigui massa alt.

Per entrar en una mica més detall, a l'hora d'especificar els casos de prova ens cal introduir la següent informació:

- **Títol:** títol auto-explicatiu, que amb una simple ullada permeti a qualsevol membre de l'equip entendre a grans trets de que va el cas de prova
- **Iteració:** en el nostre cas, en tot moment va ser SetelSys, ja que tal i com vam estar treballant al llarg del projecte no es va plantejar un treball amb diferents releases estrictes.
- **Assignació:** en el cas del projecte, en tot moment van estar assignats a mi els casos de prova, però en qualsevol altres projecte amb equips més gran és podria assignar els diferents casos de prova al seu responsable i encarregat de l'execució.
- **Estat:** el flux per defecte d'estats dels casos de prova que defineix l'eina, i que és el que hem seguit conté únicament tres estats:
  - **Design:** estat inicial i en el que romandrà el cas de prova fins que estigui totalment complert i acceptat
  - **Ready:** un cop s'ha completat el cas de prova i ja es pot procedir a la seva execució així que sigui necessari (és a dir, en teoria només s'haurien d'executar aquells casos de prova en aquest estat)
  - **Closed:** casos de prova fora d'ús.



- **Prioritat:** en tot moment, i com marca la metodologia de TMap, l'enfocament de l'estratègia de proves posa al mig i com a punt més prioritari la visió de negoci; això significa que la prioritat ens marcarà com de crític és una prova per al negoci, i per tant, en quins punts haurem d'estar més alerta.
  - **Prioritat 1:** la prioritat més alta; aquells casos de prova crítics per al negoci, que responen a les funcionalitats i seccions crítiques. En tot moment cal executar aquests casos de prova per a comprovar que no apareixen errors de regressió en aquestes funcionalitats. (per exemple, la funcionalitat de ServirPedidos dins el sistema de SetelSys és d'importància extrema per al negoci, fet que provoca que les proves que en validin la seva funcionalitat bàsica seran extremadament importants).
  - **Prioritat 2:** inclosos també en les funcionalitats bàsiques per al negoci del client; no obstant, poden pertànyer a algunes funcionalitats que poden tenir un *workaround* per a dur-se a terme. (per exemple, en alguns casos les proves del *backend*, poden passar a un segon terme, ja que pot donar-se el cas que no siguin estrictament vitals per al negoci, tot i que es faran servir de forma habitual).
  - **Prioritat 3:** casos més secundaris, tot i que valuosos per al negoci, però que en cas de detectar-s'hi algun tipus d'error no impedirien el funcionament diari de l'empresa. (per exemple, la validació de que alguns camps són obligatoris a l'hora d'afegir un nou usuari al SetelSys Runner és important, per no tenir errors inesperats però segurament que no es controli correctament aquest error no tindria un impacte crític en el negoci que podria funcionar sense problemes)
  - **Prioritat 4:** finalment, els casos menys importants des del punt de vista del negoci, ja sigui perquè es troben en àrees o funcionalitats menys rellevants, o bé perquè són situacions que potser no s'arribaran a donar mai a producció (per exemple, validacions de caire més estètic o realment en funcionalitats: que la tabulació en els formularis de donar d'alta els usuaris no estigui correctament implementada, té una importància molt baixa).

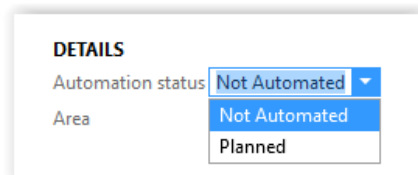




- **Estat de l'automatització:** un dels motius interessants per a seleccionar aquesta eina de gestió de proves precisament va ser el fet que permetia vincular els casos automatitzats amb l'eina, i així poder gestionar tots els resultats des del mateix punt. I aquesta integració es va poder fer tant per les proves fetes a través del Coded UI / Visual Studio, com d'aquelles fetes amb el Test Complete.

L'aplicació permet tres estats diferents:

- **Not Automated:** casos de prova que encara no s'han automatitzat.
- **Planned:** aquells casos de prova que està previst automatitzar
- **Automated:** aquells casos ja automatitzats; en aquest punt es pot veure a la pestanya d'"associated automation" quina .dll concreta correspon a l'automatització i quin cas de prova automatitzat cobreix aquest cas.



- **Steps:** un dels punts més interessants del cas de prova, ja que és amb aquesta informació que el tester encarregat d'executar la prova podrà anar seguint els passos correctes.
  - **Action:** en la primera columna cal especificar tots els passos que s'hauran d'executar per a completar la prova. En la imatge presentada a dalt, podem veure com:
    - ...
    - Clic a [Manteniment]
    - Clic a [Configurar Usuaris]
    - ...
  - **Expected Result:** és en aquesta segona columna on cal detallar quins són els resultats esperats en cada pas, per tal de que el tester pugui anar comparant a mesura que vagi executant.
    - ...
    - La pantalla de Manteniment es mostra correctament
    - ...
    - Apareix un missatge d'avís:
      - Títol: *Falta información*
      - Missatge: *Se debe anotar un nombre de usuario obligatoriamente.*

	Action	Expected Result
1.	Executar SetelSys Runner	L'aplicació s'executa correctament i apareix una finestra de login
2.	Introduir les dades correctes i clicar a [Entrar]	S'accedeix correctament i es mostra la pàgina de Procesos
3.	Clic a [Manteniment]	La pàgina de Mantenimen es mostra correctament
4.	Clic a [Configurar Usuaris]	La pantalla de gestió d'usuaris es mostra
5.	Click a [Nou Usuari]	Els camps per a crear un nou usuari s'activen
6.	Omplir tots els camps per a crear un nou usuari menys el camp del "Nom Usuari"	
7.	Clic a [Guardar]	Apareix un missatge d'avís: - Títol: <i>Falta información</i> - Missatge: <i>Se debe anotar un nombre de usuario obligatoriamente</i>

- **Altres camps:** el Microsoft Test Manager és un gestor de proves dels més potents del mercat, amb el que ofereix gran varietat de camps i possibilitats per a completar i detallar els casos de prova fins al nivell de detall desitjat. No cal entrar en detall en cadascuna de les diferents possibilitats, però si mencionar dos dels punts més interessants:
  - **Tested Backlog Items:** permet visualitzar a quins *product backlog items*, o en el nostre cas, a quins requeriments està vinculat aquest cas de prova.

STEPS

SUMMARY

TESTED BACKLOG ITEMS

LINKS

ATTACHMENTS

ASSOCIATED AUTOMATION

New

Link to

ID	Work Item Type	Title	Assigned ...	State
Tests (1 items)				
7	Product Backlog Item	Usuaris - Creació de nous usuaris	Jordi Solde...	Approved

- **Attachments:** permet incloure qualsevol informació necessària per a la prova

Finalment, com es pot veure en la imatge que es presenta a sota, a mesura que es van dissenyant casos de prova, podem gestionar-los de forma concertada dins la seva test suite, a la vegada que se'ns permet ordenar les proves segons la seva prioritat per a facilitar-ne la posterior execució

ContentsResultsProperties

Contents

NewAdd requirements

SetelSys MaidenSetelSys RunnerConfig. Recàlcul CostsConfigurar AplicatiusConfigurar BotiguesConfigurar NotificacionsConfigurar Permisos Accés DocumentsConfigurar Prioritats (botigues)Configurar Prioritats (servir comandes)Configurar TancamentsConfigurar TarifesConfigurar Usuaris7: Usuaris - Creació de nous usuaris8: Usuaris - Edició d'usuaris9: Usuaris - Eliminació d'usuarisGenerar Còpia TaulaGestió Pesos per FamíliesGrups de MagatzemsLínies Stock InexistentsMantenimentMappings Marca-ProveïdorTarifes Botigues

Test suite: 7: Usuaris - Creació de nous usuaris (Suite ID: 73) (Requirement 7)

Default configurations (1): Windows 8

OpenAddNewAssignConfigurations

Drag a column header here to group by that column.

ID	Title	Priority	Confi...	Testers	Area Path
11	Nou usuari - Nom usuari - Obligatori	3	1	Jordi Soldevila	SetelSysAutomation
12	Nou Usuari - Nom Usuari - Tamany màxim del camp	4	1	Jordi Soldevila	SetelSysAutomation
13	Nou Usuari - Nom Usuari - Nom d'usuari ja existent	3	1	Jordi Soldevila	SetelSysAutomation
14	Nou usuari - Nom complet - Obligatori	3	1	Jordi Soldevila	SetelSysAutomation
15	Nou usuari - Nom complet - Tamany màxim	4	1	Jordi Soldevila	SetelSysAutomation
16	Nou usuari - Estat - Obligatori	3	1	Jordi Soldevila	SetelSysAutomation
17	Nou usuari - Contrasenya - Obligatori	3	1	Jordi Soldevila	SetelSysAutomation
18	Nou usuari - Repetició Contrasenya - Obligatori	3	1	Jordi Soldevila	SetelSysAutomation
19	Nou usuari - Contrasenya - Longitud màxima	4	1	Jordi Soldevila	SetelSysAutomation
20	Nou usuari - Repetició Contrasenya - Longitud màxi...	4	1	Jordi Soldevila	SetelSysAutomation
21	Nou usuari - Contrasenyes - Longitud 1 caràcter	2	1	Jordi Soldevila	SetelSysAutomation
22	Nou usuari - Botigues Autoritzades - Nou usuari sens...	3	1	Jordi Soldevila	SetelSysAutomation
23	Nou usuari - Botigues Autoritzades - 1 Botiga autorit...	3	1	Jordi Soldevila	SetelSysAutomation
24	Nou usuari - Botigues Autoritzades - Diverses botigu...	2	1	Jordi Soldevila	SetelSysAutomation
25	Nou usuari - Botigues Autoritzades - Eliminar botiga...	4	1	Jordi Soldevila	SetelSysAutomation
26	Nou usuari - Botigues Autoritzades - Eliminar botiga...	3	1	Jordi Soldevila	SetelSysAutomation
27	Nou usuari - Botigues Autoritzades - Afegir botiga au...	3	1	Jordi Soldevila	SetelSysAutomation
28	Nou usuari - Màquines Autoritzades - Nou usuari sen...	3	1	Jordi Soldevila	SetelSysAutomation
29	Nou usuari - Màquines Autoritzades - Afeir màquiu...	3	1	Jordi Soldevila	SetelSysAutomation

## 8.7 Fase d'execució

Aquesta fase, i tal com marca TMap, no pot començar fins que l'objecte de prova, o una part separada del sistema que sigui provable, s'entrega. Per aquest motiu, la planificació general del projecte ja es va realitzar pensant amb diferents entregues dels diferents paquets del sistema, amb la idea de poder inicial quan abans millor com a mínim una part de l'execució de les proves.

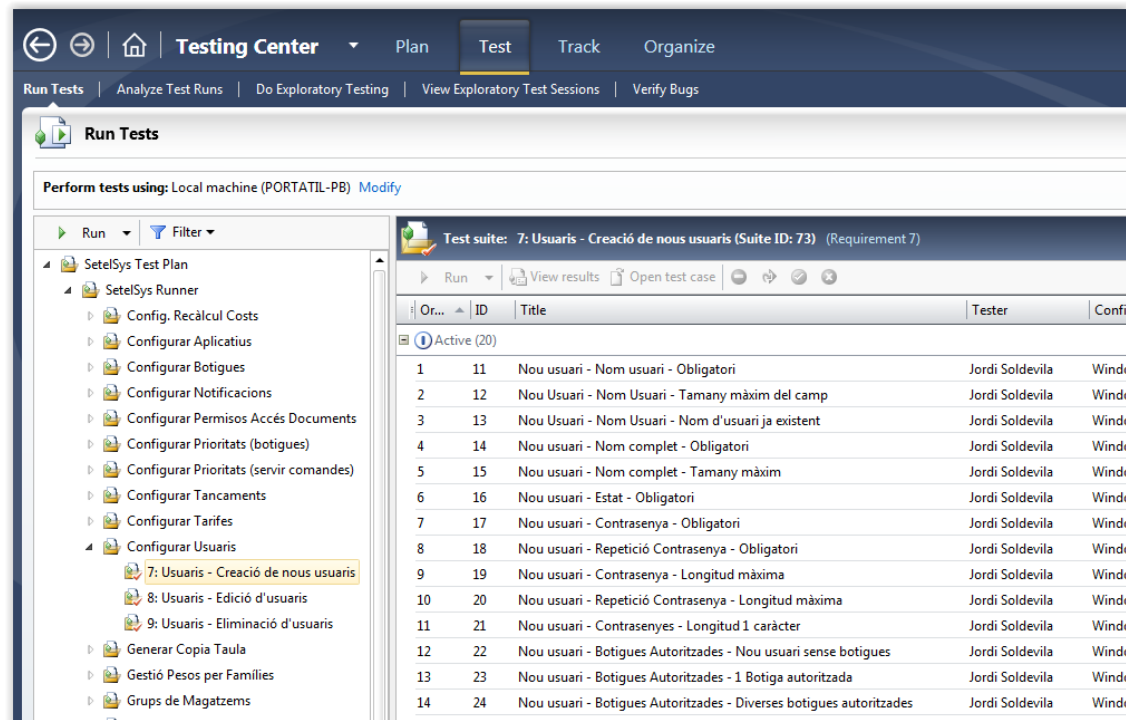
A més a més, per a començar les proves ja teníem pràcticament enllestida la fase d'Especificació, amb el que la majoria de casos de prova ja estaven correctament introduïts en el sistema de gestió de proves, el Microsoft Test Manager.

Degut ha que no és factible presentar totes les proves executades (algunes ja s'han introduït a tall d'exemple en l'apartat d'Especificació), s'ha optat per explicar breument com es procedeix a executar els casos de provar utilitzant la eina del Test Manager, per tal d'introduir també posteriorment els conceptes de gestió de defectes, i finalment poder mostrar detalls interessants en quan a creació dels defectes, el cicle de vida dels mateixos, així com una petita mostra algunes de les incidències detectades i ja solucionades, i fins i tot, alguns exemples d'errors coneguts però encara existents en el sistema, però que compleixen amb els [criteris d'entrada i sortida](#) establerts en el Pla de Proves.

Els resultats de les diferents proves realitzades, tot i anar detectant diversos errors com es veurà més endavant, van ser totalment satisfactòries, i al final de l'execució, i de la validació directament que es va realitzar en l'entorn de pre-producció va permetre afirmar que SetelSys estava preparada per a ser posada en producció.

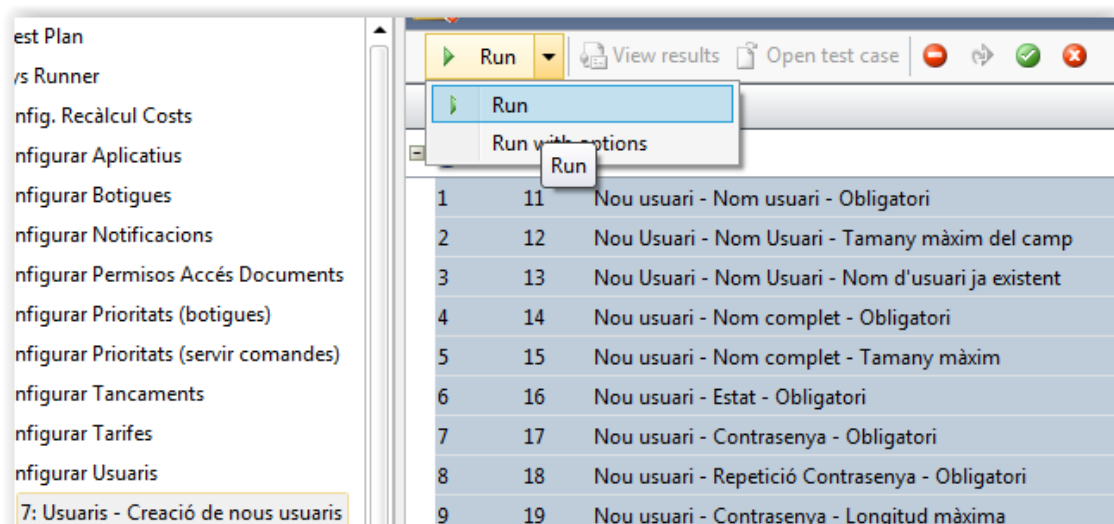
### 8.7.1 Eina de gestió de proves – Fase d'Execució

Si en la fase anterior vam acabar amb la creació dels diversos casos de prova per a cada *test suite* definida a l'abast del projecte, en aquest moment ja es va començar a treballar a la “pestanya” de test del Test Manager, on es poden visualitzar i procedir a executar els casos de prova.

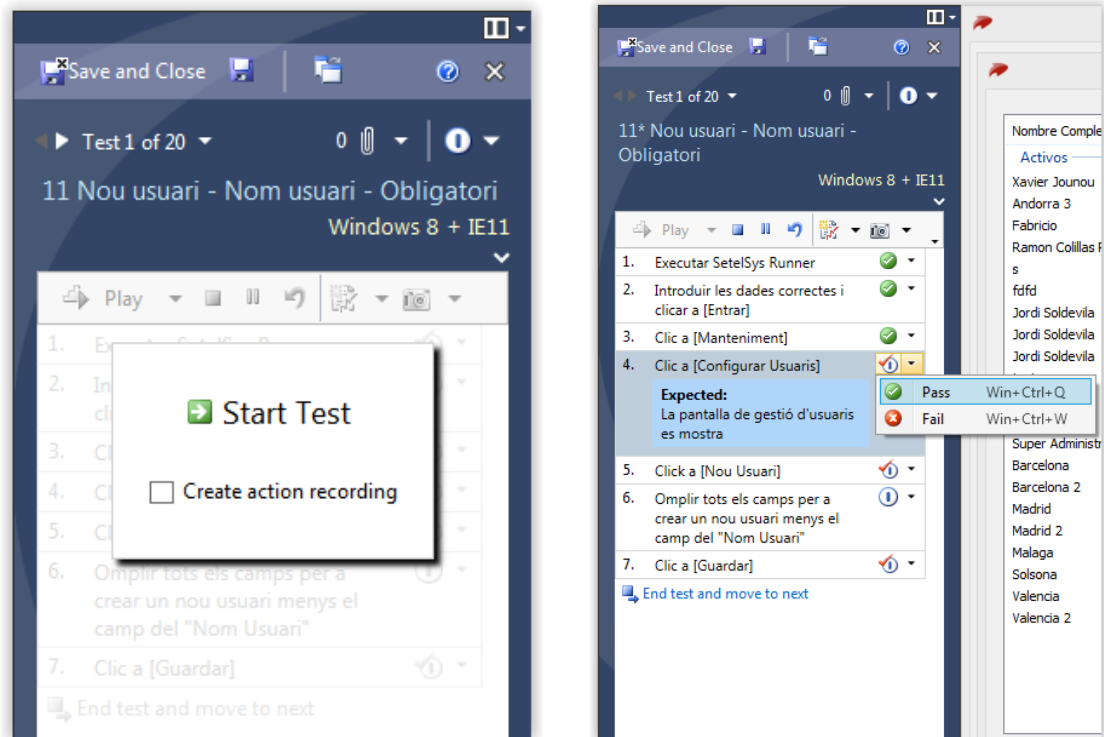


En aquesta imatge ja es pot visualitzar, que, seguint amb l'exemple de la creació d'usuaris en el SetelSys Runner, tenim els diferents casos de prova que prèviament havíem executat.

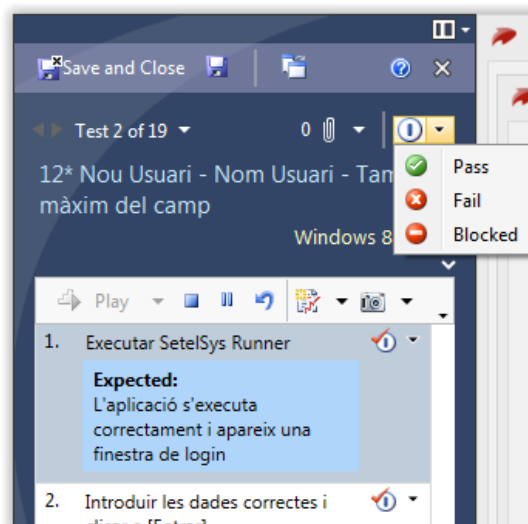
En procedir a seleccionar els casos de prova desitjats, ja es pot llançar l'execució en sí de les proves.



El Microsoft Test Manager conté un interessant “plugin” d’ajuda a l’execució, en el que es va mostrant cadascuna de les proves que s’han seleccionat, i va mostrant per un costat els passos a realitzar i per l’altre els resultats esperats:



En cada pas, es poden anar marcant els resultats (Pass o Fail), o fins i tot es pot marcar un cas de prova de forma general com a Pass, Fail o Blocked. En alguns casos, per motius derivats d’altres errors, de problemes derivats o de manca d’informació, algunes proves poden romandre com a bloquejades:



L'eina té diverses funcionalitats afegides força interessants a l'hora de desar el màxim d'informació de les proves:

- Permet adjuntar qualsevol fitxer relacionat amb la prova (per exemple, si en una prova un dels passos genera un fitxer PDF podem adjuntar-lo per a mantenir tota la informació).
- Permet realitzar captures de pantalla directament, que queden adjuntes al pas que s'estava executant en aquell moment. Aquesta funcionalitat és especialment útil en el cas d'haver de registrar algunes incidències.

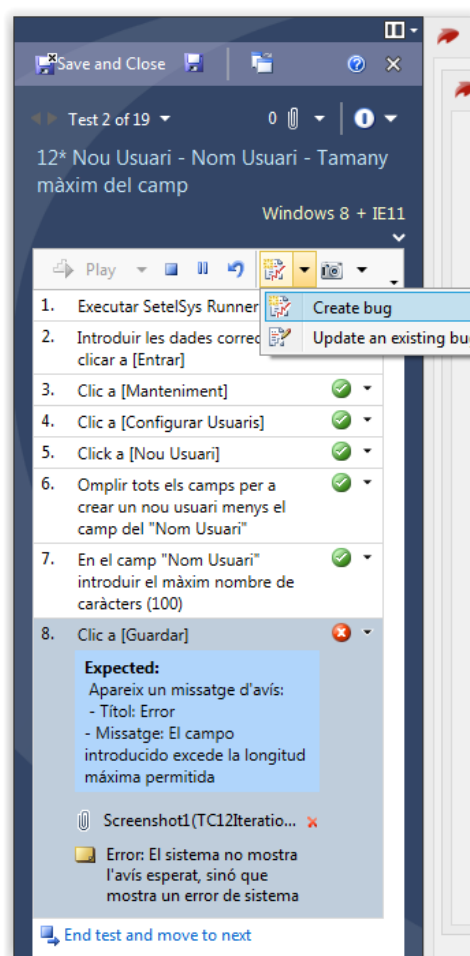
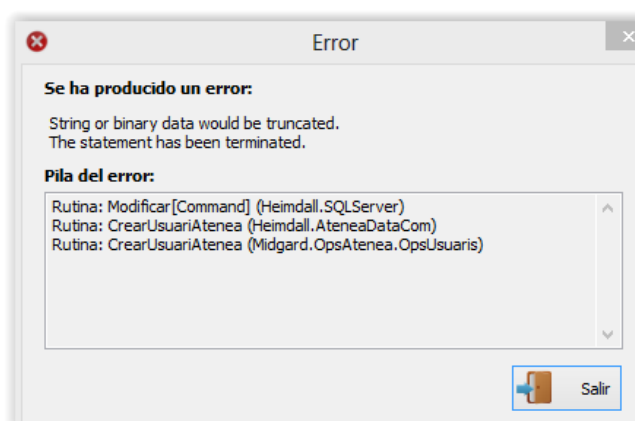
En el cas que l'execució sigui plenament satisfactòria, i un cop tots els passos hagin passat amb èxit, es podrà finalitzar la prova i seguir amb el següent.

No obstant, en el moment que detectem un error i marquem com a "fallat" un dels passos, el sistema ens permet realitzar diversos passos:

- Introduir un comentari en el pas en qüestió, on explicar l'error.
- Permet obrir directament un defecte, i com veure, i exporta tota la informació relativa a la prova directament.

Com es pot veure en la imatge, s'ha marcat l'últim pas com a "fallat", i el sistema ens ha permès introduir un comentari explicatiu de l'error.

A més, s'ha pogut realitzar directament la captura de pantalla corresponent que ha quedat adjunta al pas (Screenshot1), que adjunto també a sota. I finalment, existeix la funcionalitat per a crear directament l'error.



## 8.7.2 Gestió d'errors

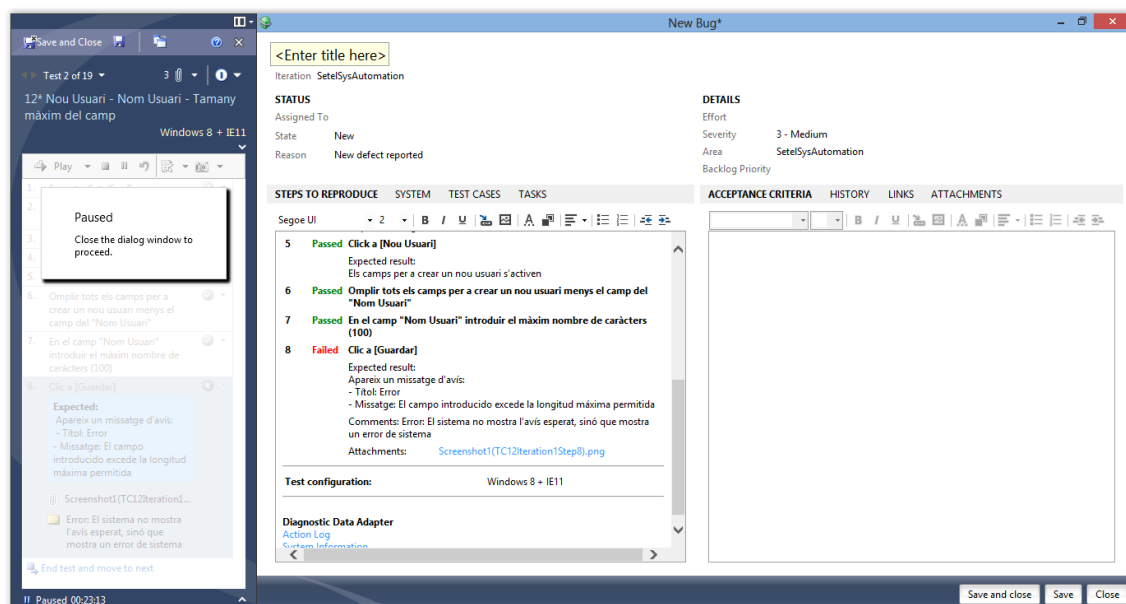
### 8.7.2.1 Eina de gestió de proves – Errors

Tal i com ja es mencionava en l'apartat anterior, el Microsoft Test Manager com a eina de gestió de proves completa també aporta la possibilitat de gestionar els errors directament a l'aplicació, fet que ens aportà diverses avantatges:

- Per un costat, la possibilitat de gestionar de forma centralitzada tots els elements que conformen el projecte simplifica el dia a dia.
- En segon lloc, al tenir tots els errors centralitzats i com veurem, registrats de les mateixes maneres per a uniformitzar-ho, va simplificar la feina del desenvolupador a l'hora de rebre i controlar les incidències, i al tester a l'hora d'anar verificant les correccions.
- I finalment, i es torna a mencionar per enèsima vegada, permet la vinculació amb els casos de prova executats, que a la vegada estan relacionats amb les suites de prova corresponent a cadascun dels requisits. En resum, traçabilitat.

### Errors durant l'execució

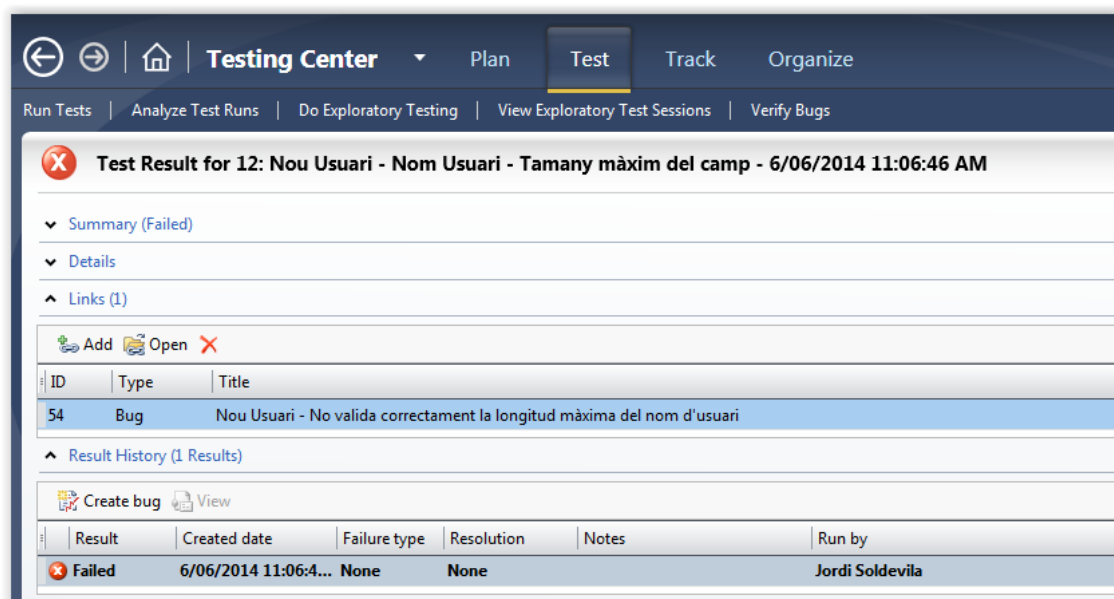
Per a seguir amb el pas anterior, quan durant l'execució trobem un error el mateix plugin del MTM permet obrir un nou defecte del sistema. En fer-ho, se'ns mostra la pàgina de creació de nou error, en la que directament ja ens ha afegit tota la informació relativa al cas de prova que s'estava executant:



El sistema directament:

- Afegeix tots els passos del cas de prova amb el seu resultat corresponent.
- En els passos on s'ha produït l'error, si s'introdueix el comentari també l'afegeix
- I també adjunta qualsevol document o captura de pantalla introduïda durant l'execució del cas de prova

A més a més, un cop desat, el sistema automàticament vincularà l'error amb el cas de prova corresponent, com es pot veure en la imatge inferior:



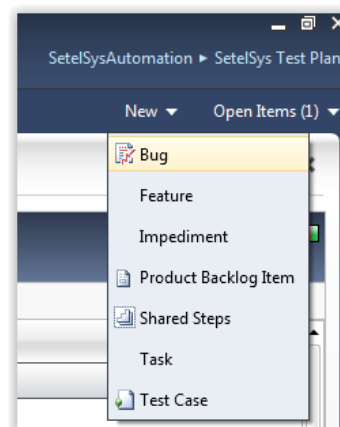
## Altres errors

A part dels errors trobats durant l'execució, en altres moments i en alguns casos de forma indirecta es van anar trobant altres errors que va caldre reportar.

L'eina permet registrar incidències directament sense necessitat de passar per l'execució de proves:

Que també ens porta a la mateix pàgina de creació que en el cas anterior, però sense cap informació ni enllaços.

Tal i com s'ha mencionat en l'apartat introductori de la [gestió de defectes](#), és molt important complimentar correctament tots els camps del defecte, per a facilitar la correcció per part del desenvolupador.

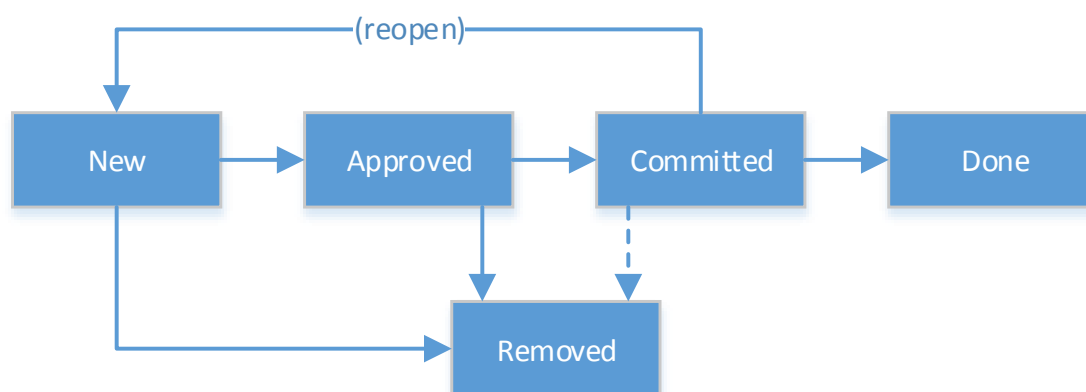




### 8.7.2.2 Cicle de vida dels errors

En tot moment durant el projecte, tot i que l'equip únicament estava format per dues persones, es va optar per a treballar com si s'estigués en un projecte més gran, amb un equip més complex. Es va optar per aquesta solució amb la idea de conèixer en detall les diferents possibilitats de les metodologies i formes de treball.

I en el cas de la gestió dels errors, en ser únicament un desenvolupador i un tester no hagués estat necessari ni crear i definir el flux de del cicle de vida dels defectes. No obstant, el flux que es va definir per a aprofundir en el coneixement del treball a través de la metodologia va ser el següent:



Els estats que s'han tingut en compte són:

- **New:** quan un nou defecte s'ha detectat en el sistema, o en el cas d'una validació no correcta, quan es reobre.
- **Approved:** quan el desenvolupador a revisat el defecte, i té suficient informació per a corregir-lo
- **Committed:** la correcció s'ha dut a terme. Pendent de validar per part de l'equip de proves.
- **Done:** la validació de la nova versió del software per a solucionar la incidència ha estat satisfactòria
- **Removed:** si en algun moment el defecte ja no té validesa. Es pot donar per diversos motius:
  - Error a l'hora d'obrir el defecte (requisits mal compresos, incorrecta execució)
  - Defecte duplicat
  - Defecte que per decisió del client no s'ha de corregir en cap moment.

### 8.7.2.3 Cicle nous casos de prova en cas d'error.

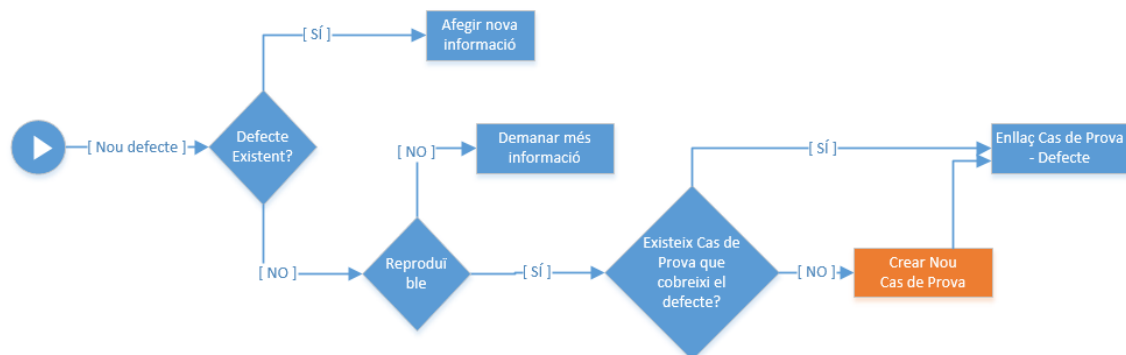
Un dels principals punts a tenir en compte és el fet que, quan es detecta un nou error, cal intentar assegurar-se que, una vegada solucionat, si pel motiu que sigui torna a aparèixer en el futur, es pugui detectar de nou sense problema.

En el cas dels errors que es detecten directament quan s'estan executant les proves, i com a resultat de les mateixes, aquesta situació ja està coberta ja que quan, en el futur, es vagi reexecutant les proves, l'error s'anirà validant.

No obstant, en alguns casos es dona la situació que l'error no es troba com a resultat directe de l'execució de proves sinó que com donar-se pel següents motius (entre altres):

- Errors col·laterals detectats durant l'execució, però no vinculats directament a l'execució.
- Durant l'execució de proves exploratòries, o durant la presa de contacte amb alguna nova funcionalitat o versió.
- Errors detectats per l'equip de desenvolupament
- Errors detectats posteriorment pel client
- Etc.

En totes aquestes circumstàncies, el flux que es va definir per a seguir va ser el següent:



Com es pot comprovar, una vegada validat que l'error era reproducible, i en cas de no existir prèviament un cas que cobrís la situació, es procedia a crear-lo, per a **incrementar la cobertura** de proves i assegurar que aquest error, en el futur, estaria cobert.

#### 8.7.2.4 Alguns dels errors detectats durant les proves.

A continuació es presenten alguns dels errors detectats en el sistema. Tal i com es va definir en els criteris d'entrada i de sortida de les fases d'especificació i execució, es podia donar la circumstància que alguns errors de severitat 3 o 4 podien romandre oberts abans de passar a producció, però en cap cas, errors de severitat 1 i 2 que podien tenir un impacte molt gran en el sistema.

Al final de l'execució i la correcció dels corresponents errors, es va assolir la fita de tenir tots els errors de nivell 1 i 2 sota control, deixant-ne alguns de menors encara oberts; però en tot moment es va informar de la situació al client.

#### Alguns dels errors corregits durant el projecte:

Més a títol de curiositat que altra cosa, ja que fora de la pròpia execució, detecció, correcció i validació no tenen molt de contingut, es mostren alguns dels errors que s'han anat corregit durant el projecte:

- Error a l'eliminar els RDPs associats a les botigues des del SetelSys Runner. Corregit procés de verificació al revisar els canvis a guardar a la base de dades perquè reflecteixi correctament els mateixos.
- Quan fa l'expedició a BCN1 dels articles de SOLSONA, han afegit 4 articles que no hi eren al document original. Si els passen els 4 articles pel lector i li donen a confirmar els surt aquest error i no els posa al traspàs, però en canvi, si els passen de un en un i van fent confirmar sí que els hi deixa. Missatge d'error: "Infracción de la restricción: PRIMARY KEY 'ALBVENTALIN\_PK'. No se puede insertar una clave duplicada en el objeto 'dbo.albventalin'. Se Terminó la instrucción."
- Si una comenda té 2 articles iguals però en 2 línies diferents, agafa el material d'una sola botiga. I si sols tenen una unitat els queda en -1. Ex: Pedido E3P 44917 , article que sols en tenen una al E11 i ha generat 2 unitats el traspas cap a Solsona ( ref S.FX.HC13-KIDCOSTA B002 talla S )
- Error de "Referencia a objeto no establecida" al expedir el material cap a SOLSONA. Quan no hi havia modificacions en l'expedició de material, intentava tancar un traspàs sobre un IDDocument no tractat. Corregit amb una condició de IsNothing.
- No poden seleccionar el text de les consultes per poder-ne fer un COPIAR del text seleccionat. Modificades les consultes perquè apareixi el text dins un TEXTBOX de només lectura, però que puguin seleccionar el text. Aleshores hauran de fer CTRL+C per a copiar el contingut.
- Segons com, el camp que s'utilitza per a marcar que el material d'una línia està reservat a SetelSys (UDMEDIDA2 dins ALBVENTALIN), es reinicialitza i fa que torni a sortir pendent de reservar. S'ha provat reordenant les línies de l'Albarà dins el Manager. En alguns casos (no es reproduïble sempre) es reinicialitza el valor.

### Algun error encara obert en el sistema:

Un parell d'errors menors en el SetelSys Runner, en l'apartat de la gestió d'usuaris. En ser el backend de l'aplicació gestionada per administradors, els errors tenien una severitat força baixa.

**Bug 31: Nou Usuari - Es pot crear un nou usuari amb un nom d'usuari ja existent**

Iteration: SetelSysAutomation

**STATUS**  
Assigned To: [empty]  
State: **Approved**  
Reason: Approved by the Product Owner

**DETAILS**  
Effort: [empty]  
Severity: **3 - Medium**  
Area: SetelSysAutomation  
Backlog Priority: [empty]

**STEPS TO REPRODUCE** SYSTEM TEST CASES TASKS

Segoe UI

1. Apareix un missatge d'error:  
- Títol: Error  
- Missatge: Este nombre de usuario ya existe en el sistema

8 **Passed** **Clic al botó de [Cancelar]**  
Expected result:  
Validar que l'usuari no s'ha creat en el sistema

9 **Failed** **Sortir de la pàgina d'usuaris, i tornar-hi a accedir**  
Expected result:  
Validar que l'usuari no s'ha creat en el sistema  
Comments: Error: el nou usuari s'ha creat

**Test configuration:** Windows 8 + IE11

**ACCEPTANCE CRITERIA** HISTORY LINKS ATTACHMENTS

Segoe UI

No s'ha de permetre la creació d'un usuari amb un nom d'usuari duplicat d'un altre.  
En mostrar l'error, no ha de proseguir amb la creació.

**Bug 54: Nou Usuari - No valida correctament la longitud màxima del nom d'usuari**

Iteration: SetelSysAutomation

**STATUS**  
Assigned To: Jordi Soldevila  
State: **New**  
Reason: New defect reported

**DETAILS**  
Effort: [empty]  
Severity: **3 - Medium**  
Area: SetelSysAutomation  
Backlog Priority: [empty]

**STEPS TO REPRODUCE** SYSTEM TEST CASES TASKS

Segoe UI

6 **Passed** **Omplir tots els camps per a crear un nou usuari menys el camp del "Nom Usuari"**

7 **Passed** **En el camp "Nom Usuari" introduir el màxim nombre de caràcters (100)**

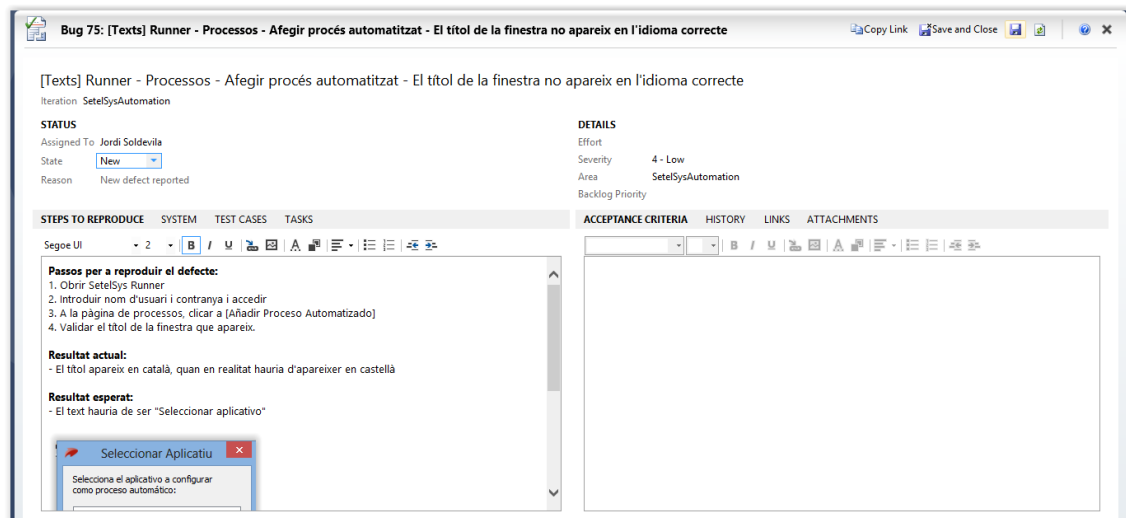
8 **Failed** **Clic a [Guardar]**  
Expected result:  
Apareix un missatge d'avís:  
- Títol: Error  
- Missatge: El campo introducido excede la longitud máxima permitida  
Comments: Error: El sistema no mostra l'avís esperat, sinó que mostra un error de sistema  
Attachments: [Screenshot1\(TC12Iteration1Step8\).png](#)

**Test configuration:** Windows 8 + IE11

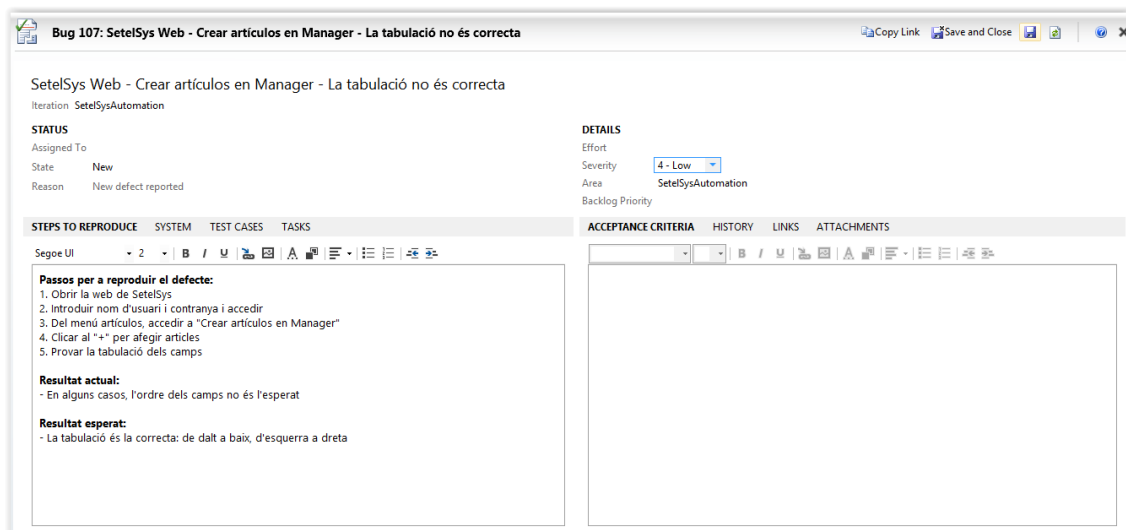
**ACCEPTANCE CRITERIA** HISTORY LINKS ATTACHMENTS

Segoe UI

Error en un altre apartat del SetelSys Runner, aquest cop un error de localització <sup>27</sup>, és a dir, de traducció dels textos de les pàgines.



O per exemple, en la web de SetelSys, un error de tabulacions en el formulari de creació d'articles al Màner, fet que es va considerar totalment menor, i que si en algun moment adquiria rellevància, es duria a terme una comprovació de tota l'aplicació en aquest aspecte.



<sup>27</sup> Wikipedia – [Language localization](#)

## 8.8 Fase de conclusió

Cap a finals del mes de maig, i durant el mes de juny es va concloure l'execució de proves del projecte de SetelSys, i per tant, es va procedir a executar la fase de Conclusió.

Segons la metodologia de TMap, hi ha dues activitats principalment en aquest fase:

- **Preservar el testware**
- **Avaluar el procés de proves**

Com en altres moments s'ha mencionat, el fet de ser un equip tant petit va impedir una fase de conclusió més potent, però que si que es va realitzar és una avaluació informal de tot el procés de testing i qualitat (alguns detalls es mencionen en les conclusions), i es va procedir a desar tot el material de test, per a futures versions del programa.

El Microsoft Test Manager té interessants funcionalitats que ens permeten reutilitzar els casos de prova dissenyats, tant per altres test suites, com per nous tests plans que vinguin en el futur: és a dir, noves versions i ampliacions de funcionalitat.

Gran part dels centenars de casos de prova preparats es podrien reutilitzar com un excel·lent joc de proves de regressió per a validar, que les noves funcionalitats, correccions, etc no tinguin impacte en les funcionalitats existents.

De cara al futur, caldria "clonar" els plans de prova que interessessin, i així mateix copiar o clonar els casos de prova interessants per a crear una regressió completa.<sup>28</sup>

---

<sup>28</sup> MSDN Blogs - [Cloning test plans using Microsoft Test Manager](#)

## 8.9 Fase de Infraestructura

Com es mencionava en un apartat anterior, segons TMap en la fase de infraestructura es tenen en compte tres elements principals:

- **Entorns de prova:** tota la infraestructura que facilita pròpiament l'execució de les proves (alguns exemples: servidors, xarxes, bases de dades, etc).
- **Eines de prova:** eines que donen suport a l'execució de les proves. Com s'ha mencionat en l'apartat 3, per al projecte s'ha seleccionat una sèrie d'eines de gestió de proves, automatització, etc, que permeten realitzar les activitats de proves més àgilment.
- **Llocs de treball:** finalment, és considera també part de la infraestructura els llocs de treball i elements necessaris per a cada treballador.

No cal estendre's més en aquest apartat, ja que en molts dels anteriors punts ja s'ha parlat de la configuració dels entorns de prova i dels diferents elements que conformen SetelSys, i també sobre les diferents eines que s'han usat per a les proves (Microsoft Test Manager, Visual Studio / TFS / Coded UI, Test Complete, etc).

En quan als llocs de treball, no s'ha cregut convenient ni tenir-ho en compte.

## 8.10 Control

Finalment, la última fase a tenir en compte en l'aplicació de la metodologia del Test Management Approach (TMap) és la fase de Control, fase que a l'igual que la de la infraestructura s'estén des del moment de finalitzar el pla de proves fins a la conclusió del projecte.

L'objectiu de les activitats d'aquesta fase és controlar i informar sobre el procés de proves d'una manera òptima, de manera que el client té una visió adequada i el control sobre el progrés i la qualitat de el procés de prova i la qualitat del sistema que s'està provant.

Com s'ha mencionat en alguns apartats, des de bon principi (ja es pot veure fins i tot en la planificació inicial que es va dur a terme), es va creure convenient una gestió molt propera amb el client (fet que TMap també posa de rellevància amb el seu Business Driven Test Management), tant a nivell de reunions per anar controlant el progrés, com per a validar i revisar qualsevol dubte, tant a nivell d'especificacions com per a tenir en compte la qualitat del producte final, i dels possibles errors que podrien restar oberts.

Inicialment es va creure interessant incorporar una sèrie de mètriques per a mesurar els diferents indicadors possibles dins les fases de proves:

- Casos de prova especificats
- Casos de prova executats
- Resultats de les execucions
- Nombre de defectes
- Nombre de defectes per prioritat o severitat
- Nombre de defectes per estat
- ...

No obstant, en molts casos les mètriques obtenen un valor afegit molt interessant quan es tracta de comparar l'evolució a través de diverses versions, per a detectar tendències i punts de millora. En tractar-se d'un projecte molt centrat a completar el producte de SetelSys, tenint en compte que com a part del Projecte de Final de Carrera únicament s'inclouria aquest producte (no s'inclouen noves possibles versions, etc), es va creure convenient desestimar la creació d'un quadre de control (Dashboard) per al projecte.

Tot i això, al llarg de tot el projecte s'ha monitoritzat constantment l'evolució de les diferents activitats per a realitzar els ajustaments pertinents per a finalitzar el projecte segons s'havia planificat.

Ja nivell més intern de l'equip, també es tenir en compte la coordinació interna i la necessitat de transmetre la situació de les àrees específiques de cadascú, i per aquest motiu des de bon principi es van convocar reunions cada dues setmanes per analitzar, debatre i plantejar alternatives el progrés del projecte.



## 9. Automatització

### 9.1 Introducció a les proves automatitzades

*"Automation is a critical component to maintaining agility, and is a priority for the entire team through established practices/disciplines and a focus on continuous improvement" [Jo Ellen West]*

En les proves de programari, l'automatització de proves és l'ús d'un programari especial (independent del programari que està en proves) per a controlar l'execució de les proves i la comparació dels resultats reals amb els resultats previstos.

Les metodologies àgils posen un èmfasi especial en l'ús de l'automatització tant com sigui possible, no obstant això, l'automatització és perfectament aplicable a qualsevol tipus de metodologia de desenvolupament o testing. L'automatització de proves pot automatitzar algunes tasques repetitives, però necessàries en un procés de prova formalitzat ja existent, o bé pot afegir proves addicionals que serien difícils de realitzar de forma manual.

Alguns dels principals avantatges de les proves automàtiques són els següents:

- Millora la qualitat: els errors humans es redueixen ja que el testing s'executa de forma sistemàtica.
- Suposa un important guany en termes econòmics i de velocitat, gràcies a l'estalvi que comporta una execució molt més ràpida.
- És repetible i reutilitzable
- Ajuda a incrementar la cobertura de proves del projecte, i per tant, que ajuda a adquirir una major confiança en la qualitat final del producte desenvolupat.
- Optimització de recursos: pot arribar a alliberar alguns testers, que poden dedicar-se a altres tasques, i per tant aconseguir un testing més intensiu.

Per altra banda, l'automatització també té una sèrie de desavantatges o punts a tenir en compte, que poden complicar un projecte d'automatització de proves:

- No es pot automatitzar tot, degut al cost que té, i en alguns casos perquè es necessària la validació humana.
- En projectes amb canvis constants, el cost de refer el codi automatitzat pot suposar una despesa de temps important.

***"It's Automation, Not Automagic!"***

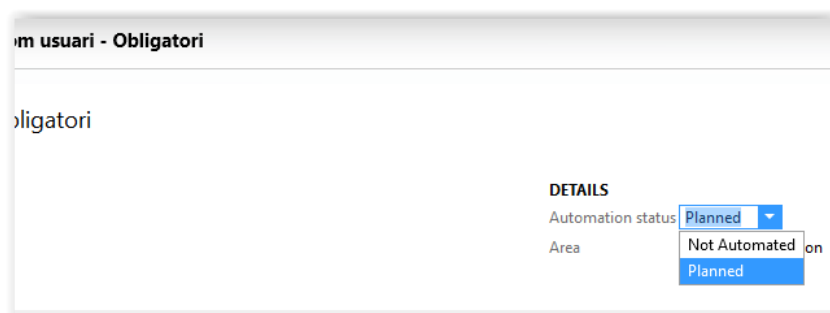
## 9.2 Estratègia de proves automàtiques i Eines

### 9.2.1 Estratègia utilitzada

Seguint dins de la metodologia de TMap, el principal punt que s'ha tingut en compte a l'hora de plantejar les proves automàtiques, en quant a la selecció de proves, és principalment la prioritat de les proves, basades en el risc que tenen per al negoci.

De les diferents proves especificades en l'eina (Test Manager), s'han anat seleccionant les proves amb les prioritats més altes.

Tal i com s'ha mostrat en el l'apartat d'especificació, l'eina MTM permet la priorització dels casos de prova. A l'hora de començar a pensar amb els casos de prova interessants per a automatitzar es va procedir a una revisió de tots aquells casos més interessants, marcant-los amb el *flag* de *Planned* en el camp de l'estat d'automatització:



A més a més, com a part de l'estratègia de proves automatitzades s'ha valorat molt el fet de minimitzar, de cara el futur, la feina de reescriure les proves; i per aquest motiu, s'ha utilitzat un patró de disseny, tal i com s'explica posteriorment.

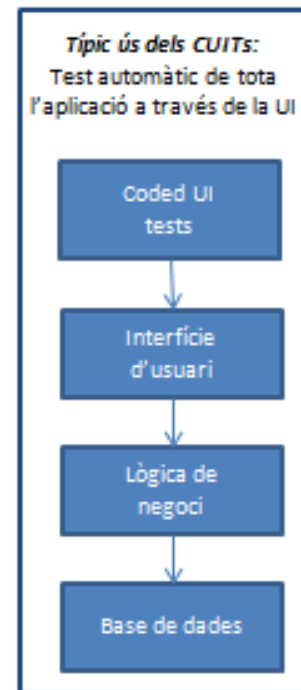
### 9.2.2 Eines utilitzades

En l'apartat 3 de Decisions Tècniques, ja es mencionava la principal característica que es va haver de tenir en compte a l'hora de seleccionar les eines adequades per l'automatització del sistema, i no és altra que el fet d'utilitzar Silverlight per a part del desenvolupament del sistema SetelSys.

Des del primer moment, vam optar per a treballar de forma integrada amb la suite **de Visual Studio**, utilitzant els **"Coded UI"**.

Les proves automatitzades que utilitzen l'aplicació a través de la seva interfície d'usuari (UI) es coneixen com a proves d'IU codificades (CUITs). Aquestes proves inclouen proves de funcionament dels controls d'interfície d'usuari. Ells permeten verificar que tota l'aplicació, incloent la interfície d'usuari, funciona correctament. Les proves automàtiques de UI són particularment útils on hi ha validació o una altra lògica en la interfície d'usuari, per exemple, en una pàgina web. També s'utilitzen sovint per automatitzar una prova manual existent.

Tal i com ja s'explicava en el perquè de la selecció, vam creure convenient no realitzar cap canvi en el propi codi simplement per adaptar-nos a aquesta circumstància, sinó que vam procedir a utilitzar una segona eina que, si que permetia el reconeixement dels elements de Silverlight.



En general el fet d'usar dues eines pot comportar una certa complexitat afegida, però gràcies a que s'integren fàcilment amb el Visual Studio i el Test Manager a través del TFS, i que afegia un punt més interessant al projecte pel fet d'adquirir més experiència en una altra eina, vam agafar-nos a l'opció de les dues.

L'eina escollida va ser el **Test Complete**, de SmartBear. Test Complete és una eina purament d'automatització, per a gestionar la creació, el manteniment i la posterior execució de les proves, tant per aplicacions d'escriptori, web com mòbil. En l'apartat corresponent veurem en detall el funcionament d'aquesta aplicació, i les modificacions necessàries per a interactuar amb els objectes.



### 9.3 Patró de disseny de proves: Page Objects Pattern

En base a l'experiència prèvia en altres projectes d'automatització, des de bon principi es va considerar el fet d'utilitzar un patró de desenvolupament per a codificar les proves automàtiques a fer per a SetelSys.

La idea d'usar un patró de disseny en les proves automàtiques és, fet i fet, en gran part pels mateixos motius que en desenvolupament s'usen patrons de disseny: una solució comú a problemes comuns, utilitzar patrons amb la idea d'evitar codi duplicat, complexitat afegida, etc.

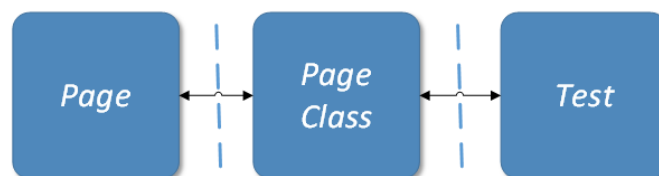
Existeixen diversos patrons de disseny que s'apliquen al desenvolupament de proves automatitzades; no obstant, el patró seleccionat per al nostre projecte és el **Page Objects Pattern**, o Patró de Pàgines d'Objectes.

#### 9.3.1 Introducció

El concepte bàsic en què es basa aquest patró és el de representar cadascuna de les pantalles que componen el lloc web o l'aplicació que ens interessa provar, com una sèrie d'objectes que encapsulen les característiques i funcionalitats representades a la pàgina. D'aquesta manera, ens permet consolidar el codi per interactuar amb els elements d'una pàgina en cadascun dels PageObjects.

En crear un PageObject, el que estem aconseguint és crear una capa d'abstracció entre el «Què podem fer / veure a la pàgina?», i el «Com» es realitza aquesta acció, simplificant enormement la creació de les nostres proves i reutilitzant el codi amb el qual interactuem amb la pàgina en concret. I alhora, qualsevol canvi que es produeixi a la Interfície d'Usuari, únicament afectarà el PageObject en qüestió, no als test ja implementats.

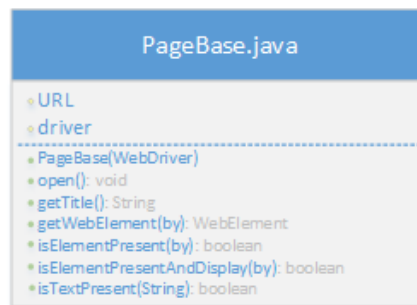
Això es deu al fet que un test mai ha de manipular directament elements de la pàgina (UI), sinó que aquest maneig s'ha de fer a través del PageObject corresponent que representa la pàgina. Per entendre'ns, el PageObject esdevé una API amb la qual fàcilment podem trobar i manipular les dades de la pàgina.



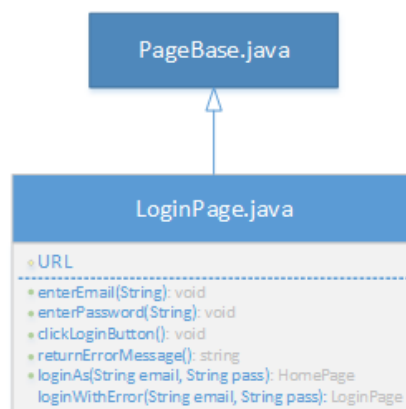
Podem trobar-nos que una pàgina correspongui a més d'un PageObject, si és el cas que algunes àrees de la pàgina són prou significatives. Per exemple, en una pàgina web, podem tenir un PageObject per al *header* i un altre per al *body*.

### 9.3.2 Detalls de la implementació del Page Objects Pattern

Una de les opcions de què disposem és crear, en primer lloc, una classe "bàsica", que posteriorment serà la que estendran cadascun dels diferents PageObjects que anem implementant. Aquesta PageBase ens aporta l'estructura bàsica i les propietats generals que utilitzarem:



I un cop creada aquesta pàgina, crear els PageObjects necessaris. Un exemple d'una pàgina de "Login" podria ser:



Lògicament, si ho creiem convenient ens podem saltar la creació de la PageBase, i crear directament els diferents PageObjects amb totes les propietats i mètodes necessaris. Com podem veure en aquest simple exemple, els mètodes que interactuen amb la pàgina en si, normalment o bé no ens retornaran cap paràmetre, o bé seran de tipus senzill, com boolean, string, etc (ex. `returnErrorMessage()` ens retorna un String amb el missatge d'error de la pàgina de login). Però quan es tracta de funcions de navegació entre pàgines, aquestes ens tornaran un nou objecte PageObject del tipus corresponent a la pàgina on naveguem. En l'exemple superior, si des de la pàgina LoginPage executem el mètode `loginas`, es realitzarà el login i conseqüentment se'ns retornarà un objecte de HomePage, que correspon al PageObject de la pàgina principal.

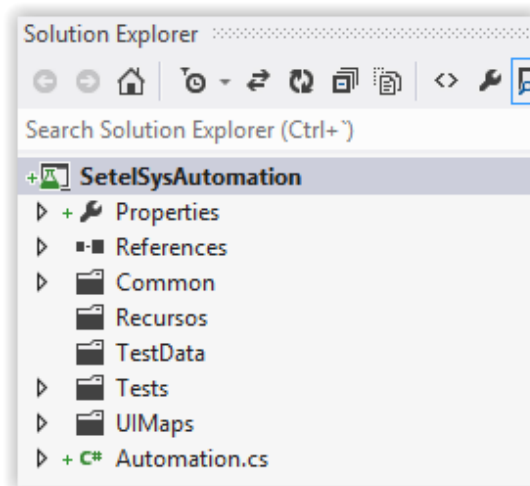
Com a conseqüència d'aquest enfocament, serà necessari crear mètodes en què l'acció a realitzar acabi amb èxit (ex: `loginas`, que ens retorna la portada), i altres en què l'acció no sigui correcta (ex: `loginWithError`, que es manté a la LoginPage).

## 9.4 Implementació de les proves – Coded UI

Començarem l'explicació i els exemples de la implementació de les proves automàtiques, comentant com s'ha enfocat el projecte de desenvolupament en el Visual Studio, usant els Coded UI.

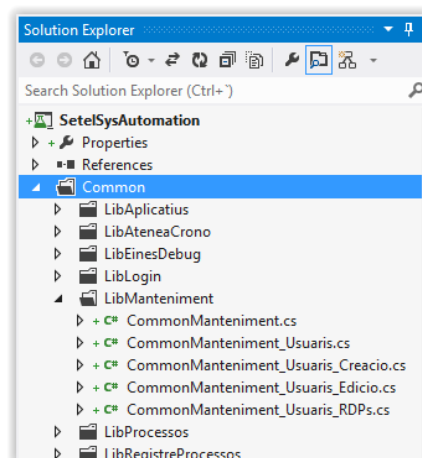
### 9.4.1 Estructura de la solució

Com apareix en la següent imatge, podem visualitzar l'estructura de la solució creada amb els diferents elements que la componen:

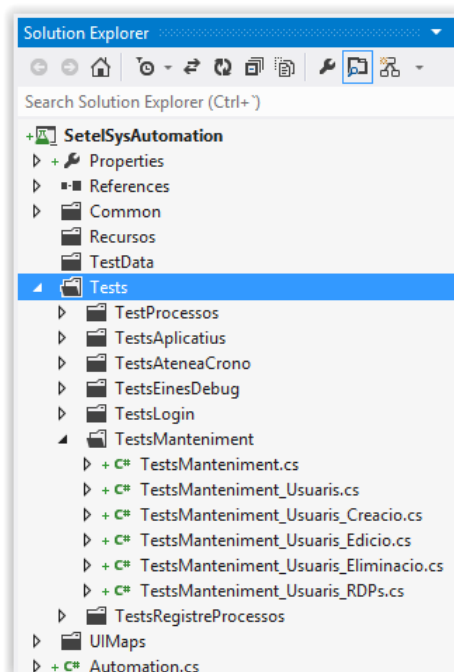


L'estructuració que s'ha seguit, i en la que s'anirà entrant en detall en cadascun dels diferents punts que es tractaran, és la següent:

- **Properties:**
  - informació i propietats
- **References:**
  - Totes les referències utilitzades en la solució
- **Common:**
  - Aquesta carpeta contindrà, en diferents sub-carpets, totes aquelles funcions “comuns” que permeten al interacció amb els objectes de la interfície d'usuari.
  - Tal i com es pot veure, les funcions comunes estan agrupades segons diferents subcarpetes, que en el fons, concorden amb les diferents “àrees” de l'aplicació SetelSys Runner.



- I dins d'aquestes àrees (per exemple, Manteniment), les funcions s'agrupen segons funcionalitats:
  - **CommonManteniment:** funcions bàsiques de l'apartat de manteniment
  - **CommonManteniment\_Usuaris\_Creacio:** funcions sobre la creació dels usuaris
  - Etc.
- **Recursos:**
  - Recursos generals necessàries per al projecte
- **TestData:**
  - Qualsevol dades de test necessàries
- **Tests:**
  - Aquesta carpeta contindrà, en diferents sub-carpets, tots els casos de prova automatitzats que cobreixen diferents àrees funcionals de l'aplicació.
  - En els casos de prova mai interactuem directament amb els objectes d'interfície d'usuari de l'aplicació; totes les accions que s'han de fer usen les funcions comunes prèviament preparades.
  - Igual que a la Common, els casos de prova s'agrupen en diferents sub-carpets relacionades amb els casos de prova especificats en l'eina Microsoft Test Manager.



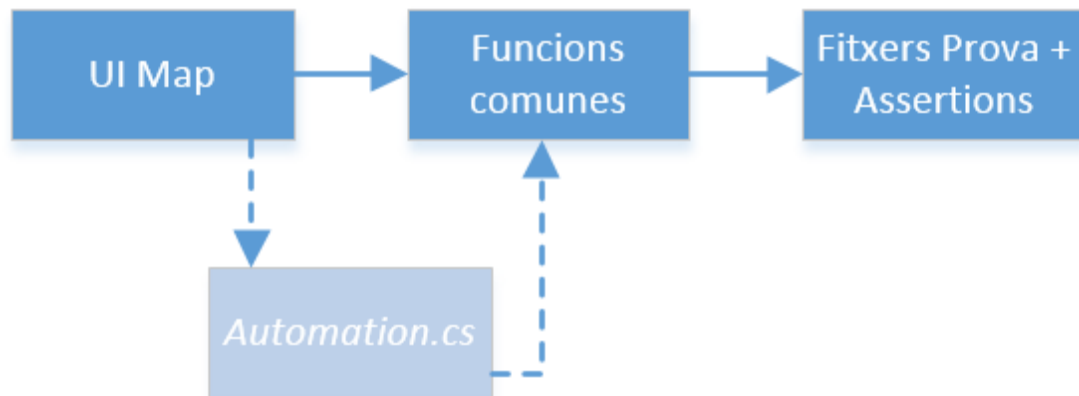
- **UIMaps:**
  - Un “UI Map” (mapa de la interfície) és un mapa dels elements de la UI.
  - En aquesta secció creem un UIMap per a cada un dels mòduls de l'aplicació que tenim sota proves.<sup>29</sup>
  - Aquests “mapes” són representacions dels diferents objectes de la UI; i els elements comuns són els que hi interactuaran.

<sup>29</sup> Per a més informació, veure [Testing a Large Application with Multiple UI Maps](#).

- **Automation.cs**
  - Fitxer principal del sistema
  - És on s'inicialitzen tots els mapes de la UI

#### 9.4.2 Procés habitual

Seguin amb el patró de disseny i l'estructura i estratègia plantejats per a treballar amb el Visual Studio / Coded UI, el procés que es seguirà normalment és el següent:





### 9.4.3 Elements de l'automatització i exemples

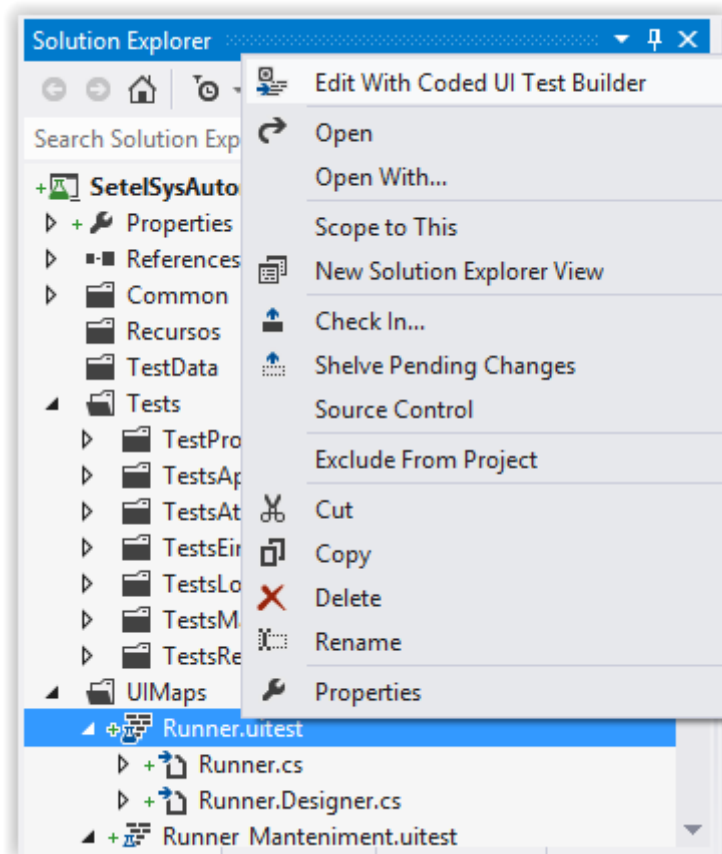
Procedim a explicar els diferents elements de la solució, i a la vegada, a anar mostrant alguns exemples:


#### 9.4.3.1 UI Maps

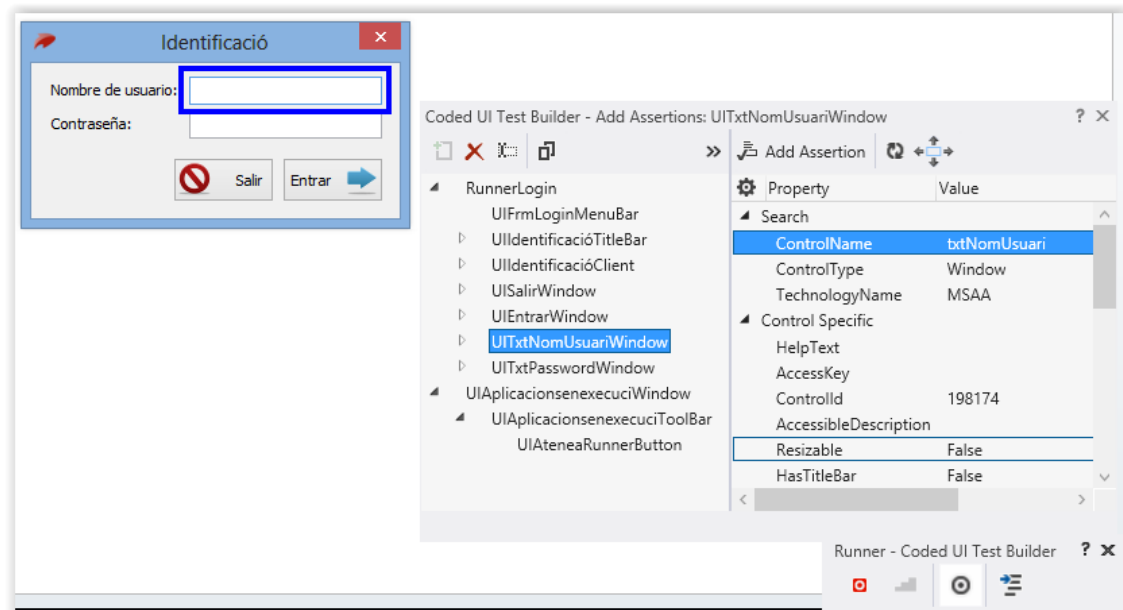
Són els mapes d'objectes que s'obtenen al processar les diferents pàgines. Tot i que es podrien crear de forma manual, la forma recomanada és a través del "Coded UI Test Builder", plugin del Visual Studio que ens permet mapejar els objectes, tot marcant-los a la interfície.

En l'exemple que s'introdueix aquí, el que s'ha realitzat és obrir un dels mapes existent, concretament el de la finestra de Login del SetelSys Runner, per a veure'n alguns detalls.

En primer lloc, cal obrir el Coded UI Test Builder, ja sigui creant un nou "Coded UI Test Map" o bé editant-ne un com es comentava:

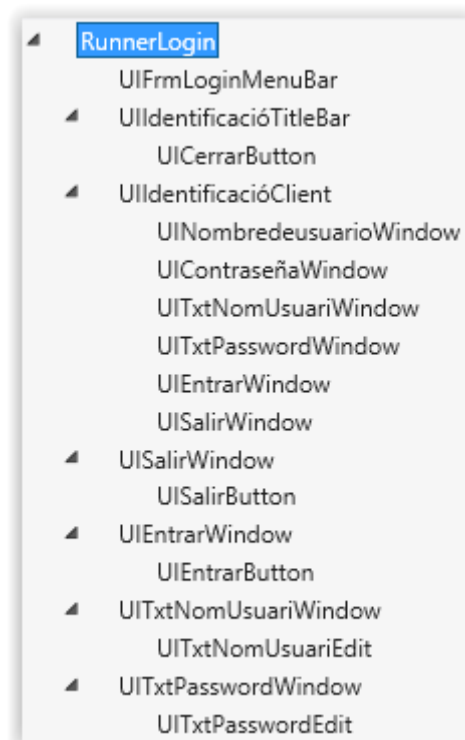


Un cop s'obre, per mitjà del selector  que permet ubicar-nos sobre els objectes i ens en mostra la informació i les propietats, talment com en les finestres de desenvolupament:



En aquesta imatge, podem veure com l'element del camp de text del nom d'usuari, queda mapejat amb el nom de "UITxtNomUsuariWindow", amb les propietats que es mostren al costat, i que seran accessibles des de les funcions comunes.

A la part esquerra podem veure l'estructura jeràrquica dels elements de la pàgina, estructura que ens serà imprescindible per a crear les funcions comunes:



- RunnerLogin fa referència a la finestra principal de la pàgina de login
- A continuació, tenim els diferents elements de la vista, com poden ser la barra de menú, barra del títol, ...
- A continuació, s'agrupa l'àrea de la identificació del client, que en un nivell inferior conté els elements de nom d'usuari i contrasenya.
- I més avall podem veure ja els botons.

#### 9.4.3.2 Automation.cs

En aquest fitxer, com s'ha comentat, és on s'inicialitzen tots els mapes de la UI, que llavors s'aniran fent servir en les funcions comunes.

És un fitxer considerablement llarg, degut a que tots els mapes cal inicialitzar-los; per a facilitar-ne l'exemple i la lectura, s'ha optat per a crear-ne un de "prova", que només conté la inicialització d'un dels mapes:

```
using SetelSysAutomation.UIMaps.Runner_Manteniment_UsuarisClasses;

namespace SetelSysAutomation
{
    public static class Automation
    {
        private static Runner_Manteniment_Usuaris usuarisUI;

        public static Runner_Manteniment_Usuaris
        Runner_Manteniment_Usuaris
        {
            get
            {
                if (usuarisUI == null)
                {
                    return new Runner_Manteniment_Usuaris();
                }
                return usuarisUI;
            }

            set { Automation.usuarisUI = value; }
        }
    }
}
```

### 9.4.3.3 Common Functions

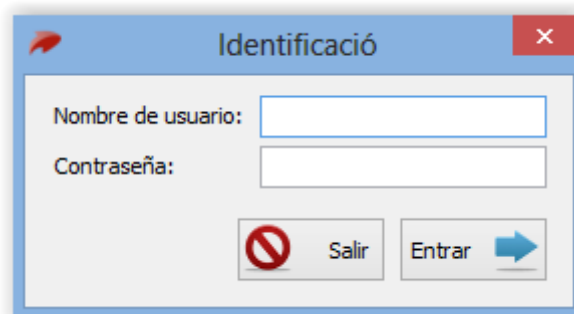
Les funcions comunes s'han d'entendre com el "pont", com la "API" entre els mapes d'objectes i els casos de prova, on realment validarem tota la informació obtinguda durant les proves.

A grans trets, les funcions comunes acostumen a ser breus funcions en les que es busca l'element de la UI amb el que interessa interactuar (per exemple, fer un clic en un botó, o introduir un text en camp text) per a procedir amb les accions. Normalment, els passos a seguir són els següents:

1. Localitzar l'objecte del sistema amb el que ens interessa interactuar
2. Realitzar alguna comprovació conforme l'element està disponible
3. Realitzar l'acció o recollir la informació que faci falta.

Com a exemples, es presenten a continuació el codi complet per interactuar amb la finestra de Login que també comentàvem en l'anterior apartat (s'agafa aquest exemple, per no estendre molt més l'explicació), i llavors també alguna funció més d'altres elements.

La finestra d'identificació al sistema és la següent:



Per tant, de forma lògica les primeres funcions que ens vindrien al cap per a interactuar amb aquest element són:

- Introducció nom usuari
- Introducció contrasenya
- Clic a Sortir
- Clic a Entrar
- Clic a "X" per a sortir
- etc

I el codi en el que consistiria aquestes funcions és el següent (separat amb diferents blocs amb les corresponents explicacions):

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Microsoft.VisualStudio.TestTools.UnitTesting.WinControls;
using System;
using System.Threading.Tasks;

namespace SetelSysAutomation.Common
{
    public static class CommonLogin
    {
```

*Nota: capçaleres de la classe; notar la incorporació de les llibreries de test del Visual Studio*

```
//Arrancar l'aplicació
public static void OpenApp()
{
    #region Variable Declarations
    string UIWpfWindowExePath = "C:\\Atenea\\SetelSysRunner.exe";
    string UIWpfWindowAlternateExePath =
"C:\\Atenea\\SetelSysRunner.exe";
    #endregion

    ApplicationUnderTest.Launch(UIWpfWindowExePath,
UIWpfWindowAlternateExePath);

    System.Threading.Thread.Sleep(2000);
}
```

*Nota: funció inicial per a arrancar l'aplicació SetelSysRunner*

```
//Clicar al botó [X]
public static void ClicX()
{
    WinButton boto =
Automation.Runner.RunnerLogin.UIIIdentificacióTitleBar.UICerrarButton;
    boto.WaitForControlReady();
    Mouse.Click(boto);
}
```

*Nota: funció per a fer clic al botó "X" de la barra de menú. Com es pot veure, en primer lloc s'assigna al botó l'element de la UI corresponent, utilitzant en aquest cas el mapa "Runner".*

```
//Clicar al botó [Sortir]
public static void ClicSortir()
{
    WinButton boto =
Automation.Runner.RunnerLogin.UISalirWindow.UISalirButton;
    boto.WaitForControlReady();
    Mouse.Click(boto);
}

//Clicar al botó [Entrar]
public static void ClicEntrar()
{
    WinButton boto =
Automation.Runner.RunnerLogin.UIEntrarWindow.UIEntrarButton;
    boto.WaitForControlReady();
    Mouse.Click(boto);
}
```

*Nota: com en l'anterior, aquí també s'assigna els botons als elements en qüestió que interessin. Posteriorment, cal esperar que el control estigui disponible per a interactuar-hi. En el supòsit d'intentar fer clic en un element que encara no estigues visible o habilitat, l'execució de l'automatització fallaria. I posteriorment ja es pot realitzar l'acció a la UI, en aquest cas com si fos un clic del ratolí.*

```
//Introduir el nom d'usuari
public static void IntroduirNomUsuari(string usuari)
{
    WinEdit camp =
Automation.Runner.RunnerLogin.UITxtNomUsuariWindow.UITxtNomUsuariEdit;
    camp.WaitForControlReady();
    Mouse.Click(camp);
    camp.Text = usuari;
}

//Introduir el password
public static void IntroduirContrasenya(string pass)
{
    WinEdit camp =
Automation.Runner.RunnerLogin.UITxtPasswordWindow.UITxtPasswordEdit;
    camp.WaitForControlReady();
    Mouse.Click(camp);
    camp.Text = pass;
}
```

*Nota: funcions per a introduir el nom d'usuari o la contrasenya*

```
//Login amb un usuari
public static void Login(string usuari, string pass)
{
    IntroduirNomUsuari(usuari);
    IntroduirContrasenya(pass);
    ClicEntrar();
}
}
```

*Nota: en alguns casos convé crear funcions auxiliars que permeten agilitzar posteriorment els casos de prova que caldria realitzar.*

Per a continuar amb algun exemple més, ens podem fixar, saltant-nos molta part del codi, en com per exemple, un cop accedíssim ja a l'aplicació de SetelSys Runner, accediríem a l'apartat de processos, i si volguéssim arribar a *Manteniment > Usuaris*, ens caldria crear la funció a **CommonProcessos.cs** per fer clic a la pestanya de "Manteniment".

```
#region ** Tabs **

//Clicar al tab de [Manteniment]
public static void ClicTabManteniment()
{
    WinTabPage tab =
Automation.Runner_Processos.UIAteneaRunnerWindow.UITabsRunnerWindow.UIMantenimientoTabPage;
    tab.WaitForControlReady();
    Mouse.Click(tab);
}

#endregion
```

*Nota: es pot veure que a grans trets la funció és igual que en el cas dels botons, però en referència a un WinTabPage.*

#### 9.4.3.4 Tests

Un cop ja es tenen totes les funcions comunes per interactuar amb els objectes de les pàgines, ja podem compondre les proves necessàries.

Hi ha una sèrie de detalls a tenir en compte:

- En primer lloc, un test no té excessiu sentit sinó procedeix a validar algun element, comportament o resultat; és a dir, les assercions (en anglès, **assertions**) ens donaran la confirmació si una prova funciona correctament o no.
- Cal incorporar l'element [CodedUITest] al principi de la classe per a marcar-la com a joc de proves
- I cal identificar les funcions com a [TestMethod] per a identificar-los com a casos de prova.

Tal i com s'ha fet per a mostrar les funcions comunes, he optat per a plasmar un exemple (que ja s'ha utilitzat en altres casos de la memòria) per a mostrar els detalls d'una implementació.

En aquest cas s'ha introduït únicament part dels casos de prova de creació de nous usuaris:

```
using System.Windows.Forms;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using SetelSysAutomation.Common;
using System.Diagnostics;

namespace SetelSysAutomation.Tests.TestsManteniment
{
    [CodedUITest]
    public class TestsManteniment_Usuaris_Creacio
    {
        public TestsManteniment_Usuaris_Creacio()
        {}
    }
}
```

*Nota: capçaleres i aparició de l'etiqueta [CodedUITest]*



```
//Test Cases Nou Usuari
#region Test Cases

[TestMethod]
public void Usuaris_AccessNouUsuari()
{
    CommonLogin.Login("admin", "admin");
    CommonProcessos.ClicTabManteniment();
    CommonManteniment.ClicConfigurarUsuaris();
    CommonManteniment_Usuaris.ClicNouUsuari();

    CommonManteniment_Usuaris.CrearNouUsuariBasic("Jordi", "Jordi
Soldevila", "Actiu", "123", "123");
    Assert.AreEqual(CommonManteniment_Usuaris.FinestraOkText(),
"Cambios guardados con éxito.");

    CommonManteniment_Usuaris.FinestraOkAcceptar();
}
#endregion
```

*Nota: aquí es pot veure un dels casos de prova, en el que es valida que es pugui crear correctament un nou usuari. Detalls interessants a tenir en compte:*

- En primer lloc, el [TestMethod] marca que la funció és un cas de prova
- En segon lloc, la idea és que els casos de prova es puguin interpretar i llegir fàcilment, i gràcies a les funcions comunes és senzill seguir el que fa la prova:
  - o A través del CommonLogin accedeix a l'aplicació
  - o Des de CommonProcessos se'n va al tab de Manteniment
  - o Allà, fer clic a l'opció de Configurar Usuaris
  - o I posteriorment a la funcionalitat de Nou Usuari.
  - o Amb una de les funcions auxiliars es procedeix a crear el nou usuari
- I finalment, es procedeix a la validació de que, en crear el nou usuari apareix el missatge de confirmació correcte.

*Lògicament, el test es podria estendre amb validacions extres, però en aquest cas s'ha optat per al camí més senzill]*

```
#region Additional test attributes

[TestInitialize()]
public void MyTestInitialize()
{
    Process[] appWindows =
Process.GetProcessesByName("SetelSysRunner");
    foreach (Process window in appWindows) { window.Kill(); }

    CommonLogin.OpenApp();
}

[TestCleanup()]
public void MyTestCleanup()
{
    //EMPTY
}

#endregion

#region Test Context
public TestContext TestContext
{
    get
    {
        return testContextInstance;
    }
    set
    {
        testContextInstance = value;
    }
}
private TestContext testContextInstance;
#endregion
}
```

*Nota: per últim, en aquesta part final s'introdueixen els conceptes de Test Initialize, que s'executarà abans de cada cas de prova, i en aquest cas ens assegurem que el procés no estigui corrent; i en cas de necessitar-ho, es pot usar el TestCleanUp per a executar-se després del cas de prova.*

## 9.5 Implementació de les proves – Test Complete

En l'apartat anterior s'ha vist com desenvolupar els test automatitzats de la part del backend, utilitzant el VisualStudio, els Coded UI.

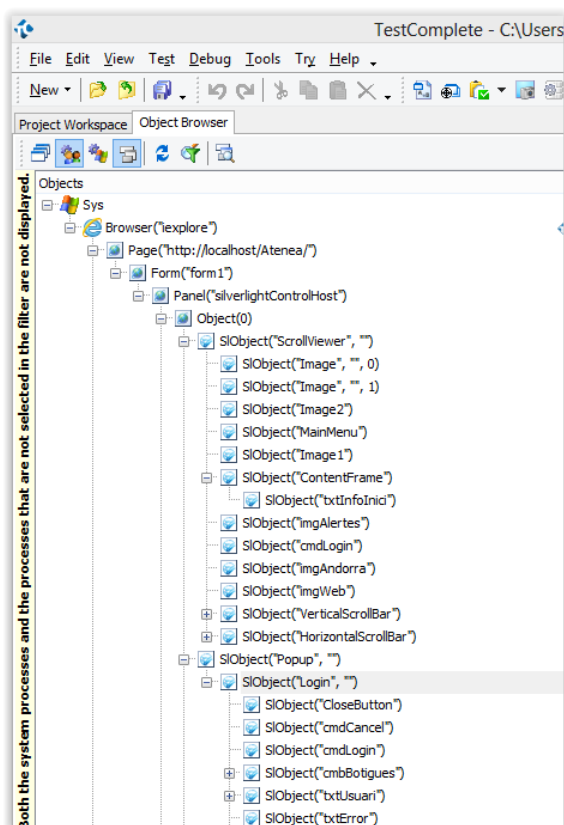
Però malauradament, en la web de Setelsys no es podia seguir amb la mateixa tecnologia i es va optar per realitzar les proves amb el Test Complete.

### 9.5.1 Automatització d'aplicacions en Silverlight

Tal i com s'especifica en la informació sobre la solució de Test Complete, sí que permet reconèixer els objectes de Silverlight de l'aplicació.<sup>30</sup> No obstant, també requereix de realitzar una petita modificació, que no és altra que modificar el fitxer .xap de la solució web en el servidor, per tal de que posteriorment es baixi de forma local en les màquines que realitzin les proves amb el Test Complete.<sup>31</sup>

La modificació consisteix en processar aquest fitxer .xap amb una utilitat pròpia del TestComplete, que és el tcAgPatcher, a través de la línia de comandes, executant la sentència:

***tcAgPatcher.exe [/silent] [/noBackup] xapfile***



A més a més d'aquest processament, hi ha una sèrie de requeriments a complir per a poder realitzar l'automatització correctament, com són l'ús dels navegadors compatibles, establint la ràtio de zoom de les pàgines al 100%.

Un cop realitzada aquesta modificació, l'eina ja és capaç de trobar els objectes de Silverlight.

Un exemple per a veure-ho és:

En el moment que sota el Panel("SilverlightControlHost") i trobem l'Object(0), si sota d'aquests dos en l'arbre apareixen significa que l'eina està reconeixent els elements.

<sup>30</sup> SmartBeard – [Testing In-Browser Silverlight Applications](#)

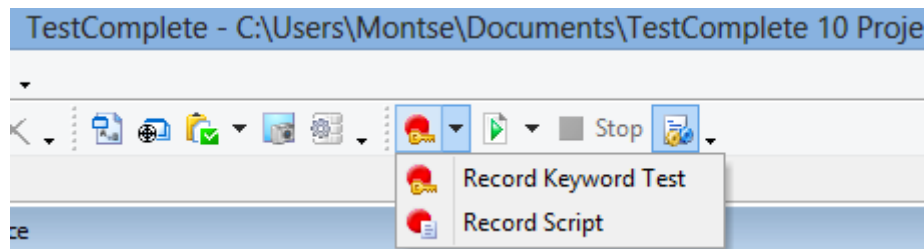
<sup>31</sup> SmartBeard – [Preparing In-Browser Silverlight Applications](#)

### 9.5.2 Opcions d'automatització de Test Complete

L'eina també ofereix la possibilitat de mapejar els diferents objectes de l'aplicació, per a utilitzar-los posteriorment en les proves.

Però a més a més, el Test Complete ens ofereix un rang de possibilitats diferents del cas previ, ja que amb aquesta eina tenim tres opcions principalment per a realitzar les proves:

1. Enregistrar els casos de prova com un **"Keyword Test"**. Un "Keyword Test" correspon a la presentació a nivell visual d'un mode d'edició basat en una graella. És el més adequat tant per als usuaris novells com per a aquells sense coneixements de programació.
2. Enregistrar els casos de prova com un **"Record Script"**. Els "Scripts" són les proves realitzades amb codi desenvolupat en un dels llenguatges de programació suportats. Pot ser més adequat per als usuaris avançats.



3. Dissenyar els casos, composant d'elements predefinits, i a la vegada usat scripts prèviament dissenyats. Pot incorporar elements i accions de Test, de Logging, de Web, pot treballar amb Checkpoints,... En realitat aquest tercer punt no és estrictament un tercer tipus de proves, sinó que usa el potencial de les diferents opcions de l'aplicació.

A continuació es mostren alguns exemples de en les tres opcions, aprofundint en alguns elements interessants a remarcar, utilitzats durant l'automatització de les proves utilitzant Test Complete.

### 9.5.3 Keyword Test

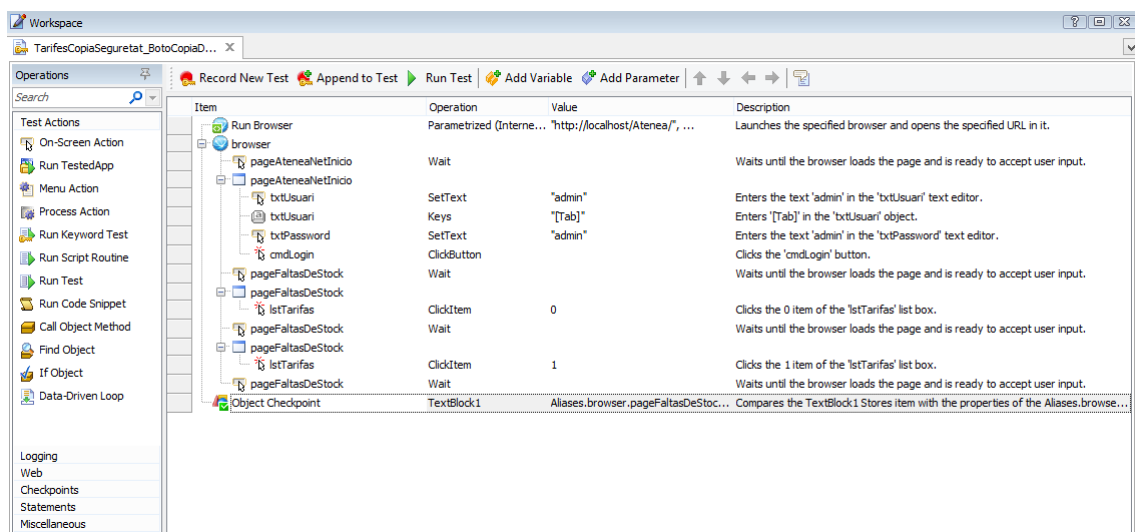
Tal i com es comentava, l'opció dels "Keyword Test" és l'opció més senzilla, que permet de forma "gràfica" l'enregistrament i posterior edició de les proves, passant a disposició de l'enginyer de proves una sèrie d'opcions a seleccionar.

Una vegada seleccionat "Keyword test" en la gravació, ens apareix el panell de control de les accions:



A partir d'aquest moment, l'eina enregistra les accions que es realitzen en l'aplicació. Per aquest exemple, s'ha optat per utilitzar una prova senzilla que simplement consistia en obrir l'aplicació, realitzar l'autenticació, i accedir al submenú de "Tarifas" > "Copia de Seguridad de Tarifas", i verificar que apareixia el botó de "Generar Copia".

El resultat de l'enregistrament és el següent:



El que es pot observar en el test enregistrat són els següents detalls:

- En primer lloc, s'usa "Run Browser", amb la URL de l'aplicació per a executar-la
- Potseriorment, ja dins el "Browser", s'accedeix a la pàgina d'inici i s'introdueixen l'usuari i el password.
- Del llistat de Tarifas, es realitza un clic a l'element 0 i posteriorment a l'element 1.
- I finalment, es realitza un "Checkpoint", una validació conforme el botó esperat existeix.

#### 9.5.4 Script Test

La segona possibilitat per a enregistrar les proves és l'ús dels "Script Test".

Per aquest exemple, la prova seleccionada és al creació d'una nova consulta per al Magatzem Central. En aquest cas, el script ha d'accedir a l'aplicació, i obrir el submenú "Logística > Consultas al Almacén Central". Un cop aquí, clicar a "Nueva Consulta" i introduir un titular de la consulta i desar-la. Finalment es procedeix a validar el missatge de correctesa.

Com es comentava al principi, aquest tipus d'script generen el codi necessari per a interactuar amb l'aplicació. El codi resultat és el següent:

```
function ConsultesMagatzemCentral_NovaConsulta()  
{  
    var browser;  
    var page;  
    var login;  
    var textBox;  
    var vobject;  
    var popup;  
    var listBox;  
    var winConsultaMagatzem;  
    var explorer;  
    var printScreen64;
```

*Nota: aquí es pot veure la declaració de la funció i les variables usades.*

```
Browsers.Item(btIExplorer).Navigate("http://localhost/SetelSys/");  
browser = Aliases.browser;  
page = browser.pageSetelSysNetInicio;  
page.Wait();
```

*Nota: S'obre el navegador, en aquest cas Internet Explorer, accedint a la web de l'aplicació (en aquest entorn, en local). I posteriorment l'aplicació "espera" a la càrrega de la pàgina.*

```
login = page.formForm1.object.Popup.Login;  
textBox = login.txtUsuari;  
textBox.SetText("admin");  
textBox.Keys("[Tab]");  
login.txtPassword.SetText("admin");  
login.cmdLogin.ClickButton();
```

*Nota: introducció de les dades d'accés, així com el clic per a entrar a l'aplicació*

```
page = browser.pageFaltasDeStock;  
page.Wait();  
vobject = page.formForm1.object;  
popup = vobject.Popup;  
listBox = popup.lstLogistica;  
listBox.ClickItem(3);  
listBox.ClickItem(7);  
page.Wait();
```

*Nota: accés al submenú de consultes al magatzem. Com es pot veure s'accedeix a "Logística", i als sub-elements especificats. També s'espera a que estigui disponible.*

```
vobject.ScrollViewer.ContentFrame.cmdNuevaConsulta.ClickButton();  
winConsultaMagatzem = popup.winConsultaMagatzem;  
winConsultaMagatzem.txtTitular.SetText("test");  
winConsultaMagatzem.cmdAbrir.ClickButton();
```

*Nota: Clic per a crear la nova consulta, introducció del titular (en aquesta prova, es va usar "test"), i clic al botó d'obrir consulta*

```
Objects.txtMissatge1.Check(Aliases.browser.pageFaltasDeStock.formForm1  
.object.Popup.winInfo.txtMissatge);  
}
```

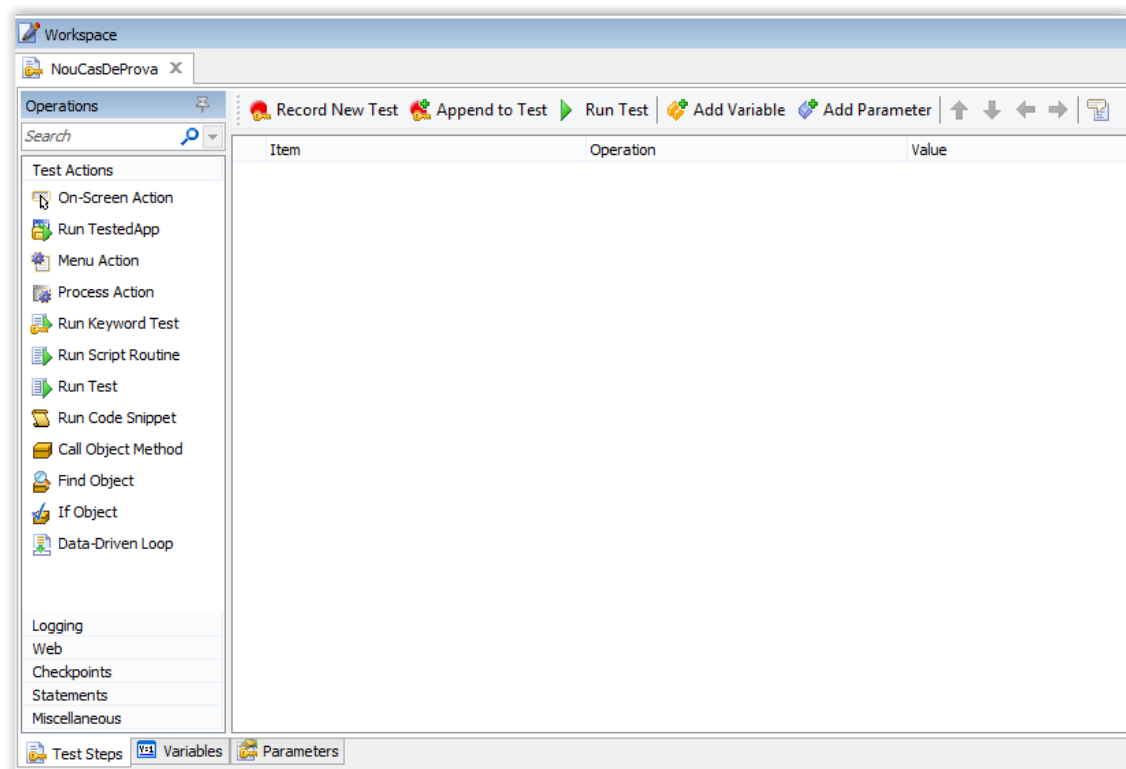
*Nota: finalment, es valida, es realitza el checkpoint de que la notificació que es mostra conté el text de generació correcta.*

### 9.5.5 Combinació d'elements

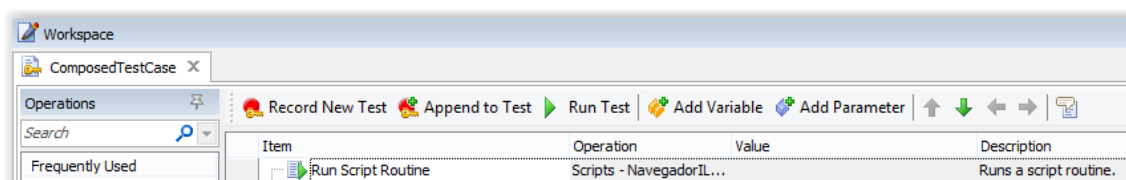
Finalment, la gràcia d'utilitzar una aplicació complexa i interessant com el Test Complete és per aprofitar tot el seu potencial, en benefici de les propies proves i de la reutilització al màxim de la feina realitzada.

En l'exemple que es mostra a continuació, en primer lloc es reutilitzar un script que s'implementa consisteix en obrir el navegador, accedir a SetelSys i realitzar l'autenticació amb un dels usuaris. Al ser una acció que pràcticament s'usava en totes les proves, es va optar per a reutilitzar els scripts. I a continuació, utilitzant les operacions que permet com a part dels "Keyword Tests", s'ha optat per a realitzar una "On-Screen Action" que consisteix en desplegar el menú de Logística.

En crear una nova prova, la finestra que se'ns mostra es la següent:



I com a primer pas, es va seleccionar utilitzar l'opció de "Run Script Routine":





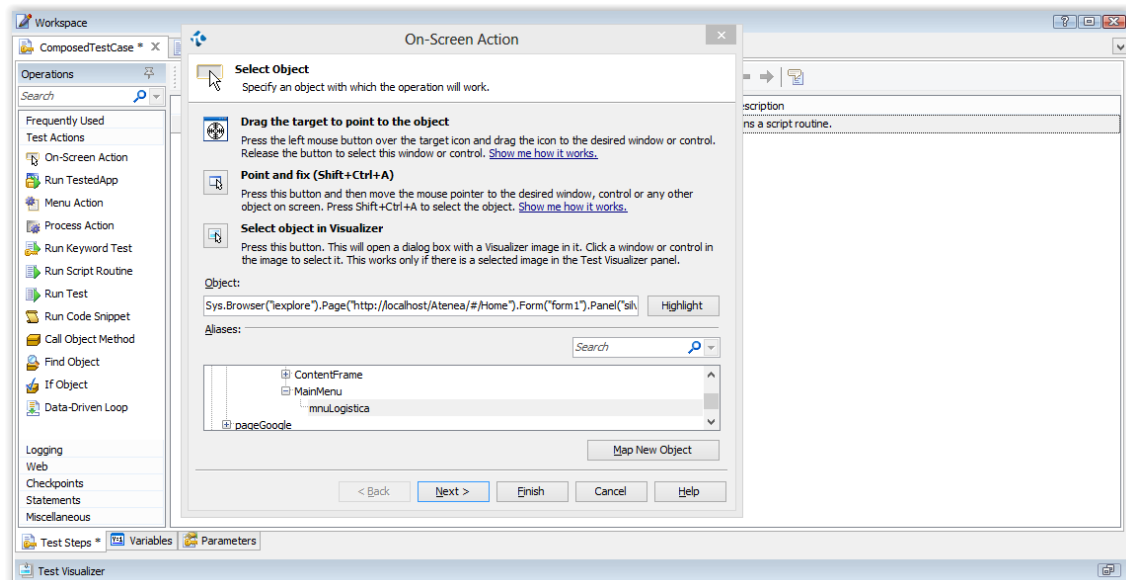
on s'ha seleccionat la rutina de "NavegadorILogin", que consisteix en el següent codi:

```
function NavegadorILogin()
{
    var browser;
    var page;
    var login;
    var textBox;
    Browsers.Item(btIExplorer).Navigate("http://localhost/SetelSys/");
    browser = Aliases.browser;
    page = browser.pageAteneaNetInicio;
    page.Wait();
    login = page.formForm1.object.Popup.Login;
    textBox = login.txtUsuari;
    textBox.SetText("admin");
    textBox.Keys("[Tab]");
    login.txtPassword.SetText("admin");
    login.cmdLogin.ClickButton();
    browser.pageFaltasDeStock.Wait();
}
```

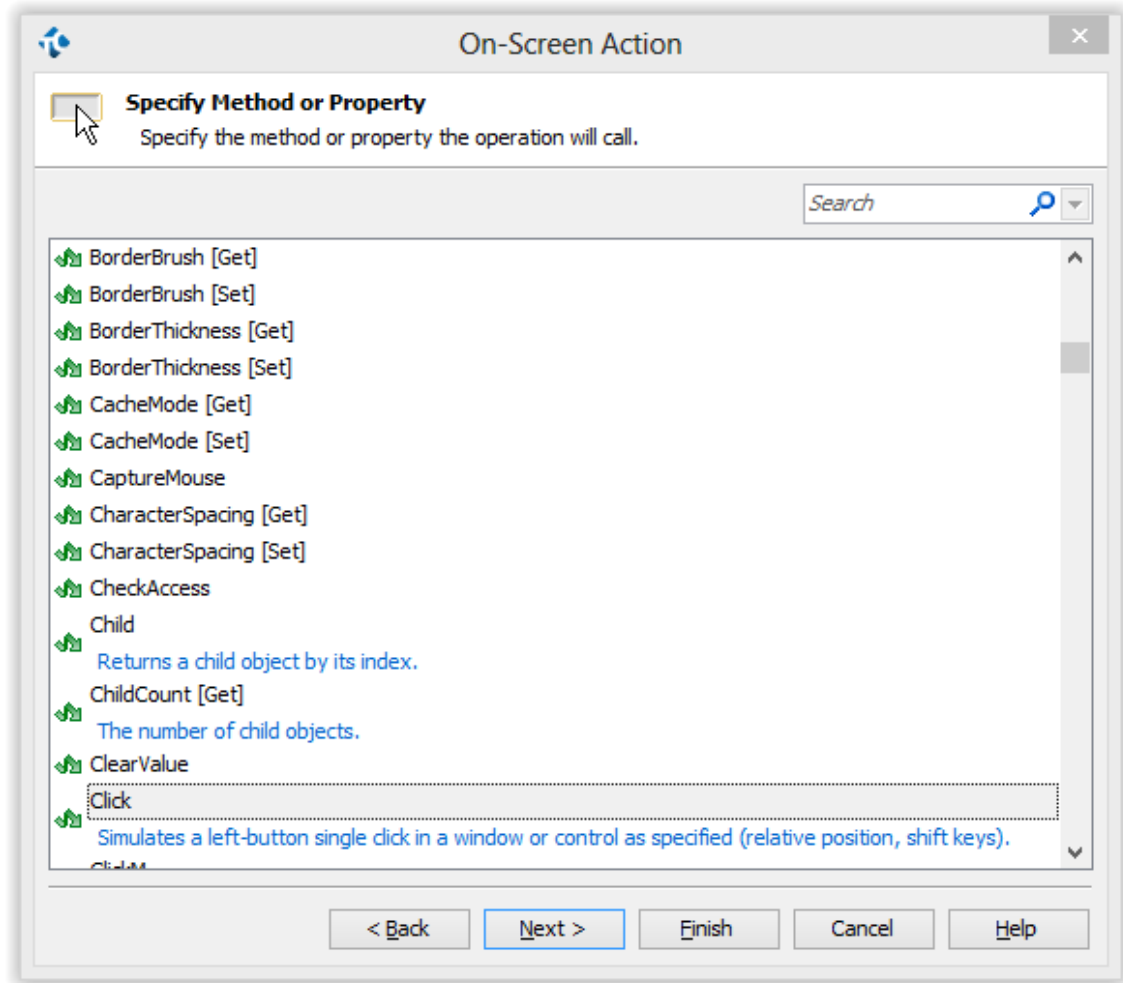
Com s'ha comentat simplement realitza l'obertura del navegador, l'accés a la pàgina de login i l'autenticació.

A continuació, s'ha seleccionat l'opció de "On-Screen Action", que permet mapejar un dels elements de la pàgina i interactuar amb ell. Per a fer-ho es realitzen els següents passos:

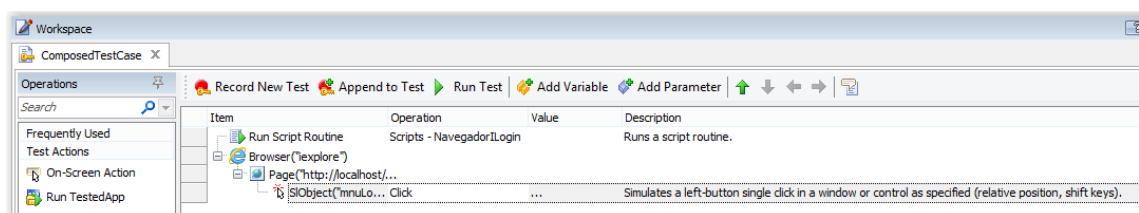
1. En primer lloc, es selecciona l'element que interessa (en aquest cas es pot veure que és el del menú de Logística):



2. I posteriorment es definexi quina és l'acció a realitzar (en aquest cas un Clic):



Quedant amb un resultat com el següent:



On tenim la combinació tant l'elements enregistrats com a scripts, com utilitzant directament objectes de la pàgina que s'està provant.

Això permet una interessant flexibilitat i aporta un potencial innegable.

## 9.6 Conclusions sobre l'automatització

El tema de l'automatització de proves és un dels grans temes de debat dins el món de Testing de Software, i en l'actualitat amb l'entrada en joc amb les metodologies àgils, l'automatització està agafant un pes cada vegada més important.

Però basant-nos exclusivament amb el projecte SetelSys i l'estratègia tant de proves com d'automatització seleccionada, algunes reflexions i conclusions interessants són les següents:

- L'automatització requereix un temps i un esforç a curt termini força importants. Per aquest motiu, l'estratègia d'automatització juntament amb l'estratègia de selecció de les proves a automatitzar és vital per al projecte.
- L'enfocament metodològic que s'ha seleccionat, amb un punt de vista de prioritització de negoci, permet rendibilitzar al màxim l'esforç que s'hi dedica.
- També és important mencionar que una eina d'automatització i un projecte d'automatització mai pot acabar de substituir la necessitat del factor humà:
  - Per un costat, perquè els enginyers de proves seran requerits per a implementar i actualitzar les pròpies proves automatitzades
  - Però per altra banda, ens molts casos podem trobar proves que no són factibles d'automatitzar o el cos de fer-ho és massa elevat.
- En el projecte SetelSys, un dels principals problemes amb el que ens hem trobat és el fet d'haver de treballar amb dues eines diferents. Per un costat, evidentment aporta un component interessant d'aprenentatge i millora de les aptituds tecnològiques. Però per la part del projecte, implica una duplictat d'esforços que en casos d'utilitzar altres tecnologies no seria necessari.
- Personalment, com a conclusió final, l'automatització ha tingut grans avantatges per al projecte, i té un gran potencial en cas de versions futures; l'ús d'una sola eina hagués ajudat a simplificar-ho, però per altra banda, l'estratègia seleccionada i l'ús de patrons de disseny en el desenvolupament de les proves ha ajudat a avançar en el projecte.

## 10. Anàlisi econòmic

A continuació es presenta la proposta econòmica desglossada en dues partides: per un costat la part de serveis, que inclou l'elaboració i implantació del projecte, i per l'altre, el hardware necessari per a habilitar l'entorn de producció.

### ***PRESSUPOST SERVEIS***

DESCRIPCIÓ	QTAT	PREU	TOTAL
<b>Desenvolupament del projecte SetelSys</b> Planificació del projecte Anàlisi de Requeriments Especificació i Disseny Tècnic Desenvolupament Testing i Automatització de Proves Finalització i Instal·lació	1487	25,00 €	37.175,00 €
<b>Descompte especial per volum (15%)</b>	1	-5.576,25 €	-5.576,25 €
<b>Obligatorietat de la contractació d'un servei de manteniment durant 24 mesos</b> <b>Cost mensual: 450€ + IVA</b>			
		<b>Total Net</b>	<b>31.598,75 €</b>
		<b>IVA 21%</b>	<b>6.635,74 €</b>
		<b>TOTAL</b>	<b>38.234,49 €</b>

A nivell de serveis hem elaborat una proposta en la que establim una tarifa de 25,00€ + IVA per hora de desenvolupament. En el còmput d'hores hem afegit un 10% de desviació per demores i imprevistos que es puguin produir. Partint de 1352h, afegint aquesta desviació ens queda un total d'hores de 1487h.

Per tal de poder competir amb propostes alternatives de servei de possibles competidors, hem ofert al client, un descompte especial per volum de un 15% sobre l'import base del projecte pel que fa a la part de serveis.

També s'ha vinculat a la proposta, un contracte de manteniment vigent amb un mínim de 24 mesos, amb renovació opcional, per un import de 450,0€ + IVA per mes.

## ***PRESSUPOST HARDWARE***

DESCRIPCIÓ	QTAT	PREU	TOTAL
<b>Server HP Proliant DL360P Gen8</b> Xeon E5-2620 (2.0/12), 8Gb, Controladora SAS HP P420i, DVD	1	1.965,00 €	1.965,00 €
<b>Disc dur hot plug SAS 15.000rpm 500GB Gen8 (6GB)</b>	3	289,00 €	867,00 €
<b>MS-Windows2008 Server Standard. Server + 5CAL</b>	1	535,00 €	535,00 €
<b>Instal·lació i configuració del servidor</b>	1	350,00 €	350,00 €
<b>Opcions</b>			
<b>Mòdul memòria RAM ECC HP 8GB (1 x 8GB)</b>	1	185,00 €	185,00 €
<b>Font d'alimentació redundant HP</b>	1	250,00 €	250,00 €
<b>Extensió de garantia HP CarePack - 3 anys</b>	1	325,25 €	325,25 €
		<b>Total Net</b>	<b>4.477,25 €</b>
		<b>IVA 21%</b>	<b>940,22 €</b>
		<b>TOTAL</b>	<b>5.417,47 €</b>

A nivell de hardware s'ha ofert tots els components i elements opcionals que hem descrit i analitzat en la corresponent [secció del document](#).

## 11. Conclusions

En tota metodologia tant a nivell de desenvolupament com de testing, s'acostuma a finalitzar per una fase de revisió i millora del procés, i no voldríem deixar passar l'oportunitat de realitzar aquesta tasca.

### *Conclusions a nivell de l'aplicació SetelSys:*

Estem molt contents de poder dir que s'ha aconseguit arribar a la satisfacció de la major part dels objectius plantejats a l'inici del projecte. Considerem oportú mencionar-ne els següents:

- Aconseguir unificar un sistema heterogeni, fragmentat i disseminat, plantejant el nou sistema en base d'una arquitectura més coherent i racionalitzada, adaptada a les necessitats reals del dia a dia de l'empresa.
- Les situacions actuals del mercat requereixen d'un desplegament de noves funcionalitats ràpid i eficient, responen a les necessitats canviants de l'empresa. A través de SetelSys s'ha millorat, de forma destacada, el temps en la implantació efectiva d'aquestes noves funcionalitats.
- S'ha assolit l'objectiu de crear un sistema suficientment configurable i senzill en el seu ús com per haver reduït la dependència d'un perfil molt tècnic i especialitzat, dotant al sistema d'una alta interactivitat i parametrització.
- Des de bon principi, l'equip de treball ha tingut clara la importància de mantenir una consciència de qualitat i aplicar amb rigor les normes seleccionades. L'aplicació de metodologies estructurades de proves ha repercutit en un increment de la qualitat final del sistema en termes de: usabilitat, robustesa, fiabilitat i rendiment.
- Hem dotat al sistema d'un motor capaç de gestionar els errors produïts en l'ús de les diferents aplicacions del sistema, provinents de situacions inesperades que no han estat validades o verificades en els processos de desenvolupament i testing.
- La implementació de les polítiques de seguretat corresponents en l'arquitectura del sistema s'ha traduït en un augment de la seguretat i l'accés controlat.
- S'ha arribat a un nivell de millora del rendiment de la majoria d'aplicacions tant en el temps de còmput com en els resultats gràcies a l'ús d'eines i mecanismes més adequats als processos de negoci de l'empresa i la complexitat algorísmica d'aquests.
- El fet d'haver analitzat en detall tots els processos de l'empresa durant l'anàlisi de requeriments, ha permès a la pròpia empresa detectar i requerir millores en el seu flux de treball.

No obstant hem detectat diferents elements que tot i ser plantejats en una proposta inicial d'objectius, no queden del tot assolits o dibuixen certes mancances:

- La manca del *feedback* adequat amb l'empresa desenvolupadora de la solució ERP i les seves bases de dades de treball, poden causar possibles escenaris d'incoherència en els resultats de certes funcionalitats. Tot i desenvolupar eines per la detecció precoç d'aquestes possibles situacions i establir uns protocols de comunicació interna prèvia instal·lació de qualsevol actualització, no es pot garantir en tot moment la correctesa de la comunicació entre ambdues plataformes.
- La nostra ambició en la creació d'un sistema de control d'errors potser ha estat excessiva, resultant en un sistema sobredimensionat i realment no aprofitat en la gestió d'incidències o detecció d'anomalies.
- Tot i que s'ha mencionat la millora de rendiment global de l'aplicació, per a realitzar un punt d'auto crítica seria interessant mencionar que algunes de les funcionalitats existents requeririen d'un treball de millora i increment del seu rendiment.

### *Conclusions a nivell d'eines de treball*

Ens hem sentit còmodes en la utilització de les diferents eines i solucions de les plataformes de Microsoft, tant a nivell de desenvolupament com de testing, ja que tant per experiències laborals prèvies com per facilitat d'ús, ens ha permès utilitzar-lo amb agilitat.

No obstant i seguint amb el punt d'autocrítica, trobem oportú mencionar que la selecció de Silverlight com a mitjà d'interacció amb l'usuari seria un dels elements que possiblement ens replantejaríem. Tot i que ha aconseguit complir amb les necessitats del projecte i l'entorn, la falta de continuïtat d'aquesta tecnologia pot suposar limitacions en un futur i per altra banda ha implicat l'ús de dues eines d'automatització proves que haurien pogut resumir-se en una.

### *Conclusions a nivell de projecte de final de carrera*

Ja per a finalitzar, no volem passar sense mencionar un parell d'elements que ens han semblat rellevants en quan al projecte de final de carrera:

- Trobem molt interessant el fet de treballar en equip i no a nivell individual ja que aporta un punt de realitat al projecte, que en el dia laboral tots ens trobem.
- El fet de ser un projecte amb un fonament real ens ha permès enfrontar-nos a les dificultats presents en el mercat actual i tant dinàmic. Això ha suposat una sèrie de dificultats afegides que d'altra banda donen un valor molt més alt a la feina realitzada.

### 11.1 Possibles millores futures

Al llarg del projecte i posteriorment amb la implantació, s'han detectat una sèrie de possibles millores tant a nivell d'optimització com d'extensió de la plataforma. Les més rellevants a destacar són les següents:

- Tot i no estar en el full de ruta del projecte, ni en les necessitats actuals del client, s'ha pogut observar els possibles avantatges que suposaria la extensió del sistema en plataformes mòbils. Inclús l'especialització de certes funcionalitats dirigides a l'ús massiu en les botigues i magatzems del client.
- Caldria replantejar alguns elements arquitectònics que dotessin al sistema d'un major rendiment en la carrega inicial de la interfície, per a donar resposta a les mancances pel que fa a connectivitat existents a l'empresa. Al no ser un factor sota el nostre control, caldria adaptar-s'hi.
- Optimitzar el grau d'automatització en el desplegament de noves versions en l'entorn de producció.
- Caldria millorar el control intern en l'execució de processos crítics de llarga duració, per evitar pèrdues innecessàries de temps al re-executar aquest passos i minimitzar l'impacte en els fluxos de treball.
- En certs processos s'ha detectat reiteradament necessitats de parametrització que caldria cobrir per reduir la dependència d'una actuació tècnica.



## 12. Bibliografia i referències

### *Planificació i gestió del projecte*

- [Microsoft Project](#)
- [Toggl](#)

### *Desenvolupament*

- **ASP.NET**
  - Microsoft – [The official Microsoft ASP.NET site](#)
  - ASP.NET – [ASP .NET Web Forms](#)
  - Code Digest – [HttpHandler in ASP.Net](#)
  - MSDN – [ASP .NET Web Configuration Guidelines](#)
- **MVC**
  - Microsoft – [ASP MVC Overview](#)
  - Wikipedia – [Model View Controller](#)
- **Silverlight**
  - Microsoft – [Que és Silverlight?](#)
  - Wikipedia – [Format d'arxius .XAP](#)
  - MSDN – [What is XAML?](#)
  - Tech Republic – [Create rich interfaces with Microsoft Silverlight](#)
  - MSDN – [Create a new Silverlight Project](#)
  - Code Project – [Using WCF Service with Silverlight](#)
  - InformIT – [Silverlight Best Practices: Asynchronous Programming](#)
  - MSDN – [Navigation Overview in Silverlight](#)
  - MSDN – [Silverlight Controls](#)
  - Microsoft – [Microsoft Silverlight 4 Offline Documentation](#)
  - MSDN – [Hosting and consuming WCF Services](#)
- **MVVM**
  - MSDN – [Design Patterns - MVVM - Model View ViewModel Pattern](#)
  - Wikipedia – [Model View ViewModel](#)
- **WPF**
  - Wikipedia – [Windows Presentation Foundation](#)
- **Windows Services**
  - MSDN – [Introduction to Windows Service Applications](#)
  - MSDN – [Create Windows Services](#)

- **Visual Studio i Visual Basic .NET**

- Microsoft Visual Studio – [Pàgina de presentació de Visual Studio](#)
- MSDN – [Windows Forms](#)
- Tutorials Point – [VB.Net – Basic Controls](#)
- MSDN – [Expression Blend for Visual Studio](#)
- MSDN – [How Visual Basic Provides Polymorphism](#)
- MSDN – [Try/Catch Syntax in VB .NET](#)
- MSDN – [Serialization \(C# and Visual Basic\)](#)
- MSDN – [Inheritance in VB .NET](#)
- MSDN – [Parsing XML using XmlReader Class](#)
- MSDN – [Handling and Raising Events in .NET](#)
- MSDN – [Creating Windows Form User Control](#)
- DotNetHeaven – [Start or Stop a Windows Service using VB.NET](#)
- MSDN – [SqlCommand Class Reference](#)

- **IIS**

- IIS.Net – [Microsoft IIS \(Internet Information Services\)](#)

- **Microsoft SQL Server**

- MSDN – [SQL Server Management Studio](#)
- MSDN – [Microsoft SQL Server 2008 Editions](#)
- MSDN – [Information Schema Views using Transact-SQL](#)

- **Microsoft Visio**

- Microsoft Office – [Microsoft Visio 2010](#)
- Microsoft Office – [Reverse engineer an existing database using Visio](#)

- **Generació d'informes**

- MSDN – [Microsoft Reports](#)
- SAP – [Crystal Decisions Crystal Reports](#)

- **Altres**

- 7Zip.ORG – [Pàgina web oficial de 7Zip](#)
- MSDN – [RDP \(Remote Desktop Protocol\)](#)
- Google Sites – [Algorisme de Levenshtein](#)
- Wikipedia – [Operativa CRUD](#)

*Eines de gestió de codi*

- [Apache Subversion](#)
- Tortoise SVN - [Home](#)
- AnkhSVN – [Subversion Support for Visual Studio](#)

## Testing

- **Testing:**
  - Llibre: The Art of Software Testing. Glenford J Myers, 1979
  - Llibre: TMap® NEXT for result-driven testing. Koomen, Tim; van der Aalst, Leo; Broekman, Bart; Vroon, Michiel. 2006
  - The Turing Digital Archive – [Checking a large routine](#)
  - Abstract and Bibliography from NBS Special Publication 500-106, "[Guidance on Software Maintenance](#)"
  - US Department of Transportation - [Developing Functional Requirements](#)
  - Wikipedia – [Language localisation](#)
- **Test Management Approach (TMap)**
  - SOGETI – [TMap Webpage](#)
  - TMap NET - [Downloads](#)
  - TMap NEXT Downloads – [Template Test Plan System and Acceptance Tests](#)
- **Microsoft Test Manager:**
  - [Testing your application using Microsoft Test Manager](#)
  - MSDN Blogs - [Cloning test plans using Microsoft Test Manager](#)
- **Team Foundation Server (TFS):**
  - Visual Studio - [TFS Overview](#)
  - Microsoft Developer Network – [Team Foundation Server](#)
- **Testing de solucions Silverlight:**
  - Microsoft Developer Network - [Testing Silverlight Applications with Coded UI Tests or Action Recordings](#)
- **Automatització:**
  - Selenium Wiki - [PageObjects](#)
  - Microsoft Developer Network - [¿Qué es un patrón de diseño?](#)
- **Automatització amb Coded UI:**
  - MSDN - [Testing a Large Application with Multiple UI Maps.](#)
- **Test Complete:**
  - SmartBear – [Automated Testing](#)
  - Test Complete – [Integration With Visual Studio](#)
  - SmartBear – [Testing In-Browser Silverlight Applications](#)
  - SmartBear – [Preparing In-Browser Silverlight Applications](#)